Building Streaming Data Pipelines with Elasticsearch, Apache Kafka, and KSQL

London Elastic Meetup, 31 Jan 2018

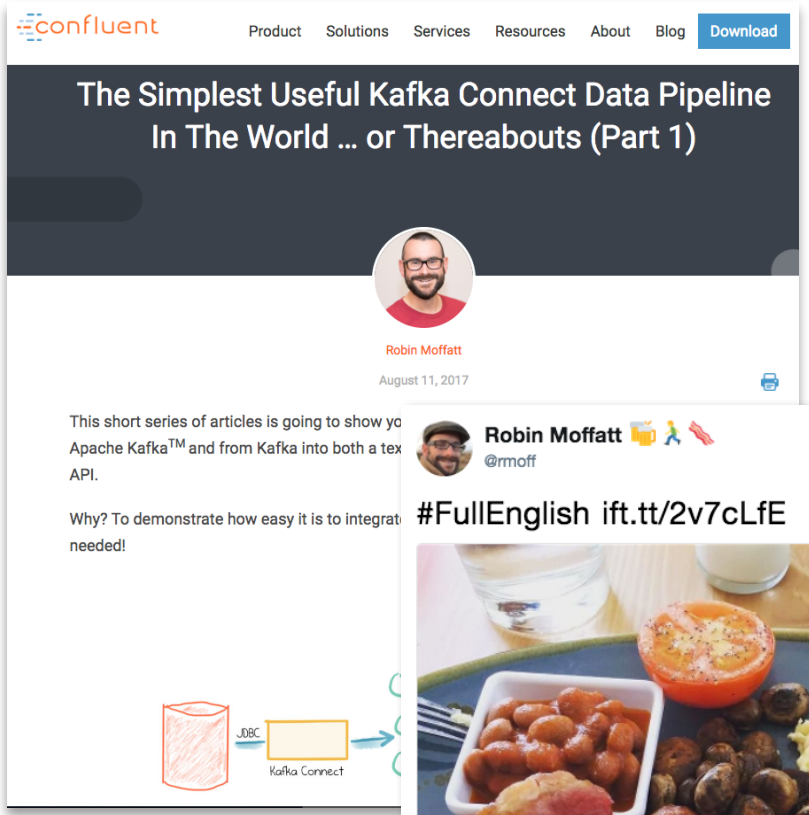
Robin Moffatt, Partner Technology Evangelist, EMEA

@rmoff robin@confluent.io

https://speakerdeck.com/rmoff

$ whoami

- Partner Technology Evangelist @ Confluent
- Working in data & analytics since 2001
- Oracle ACE Director
- Blogging : http://rmoff.net & https://www.confluent.io/blog/author/robin/
- Twitter: @rmoff
  - Geek stuff
  - Beer & Fried Breakfasts



I ❤️ Elastic 😄

elastic Products Cloud Services Customers Learn download

Blog News Engineering User Stories Releases

09 MARCH 2016 USER STORIES

Visualising Oracle Performance Data with the Elastic Stack

By Robin Moffatt

Share

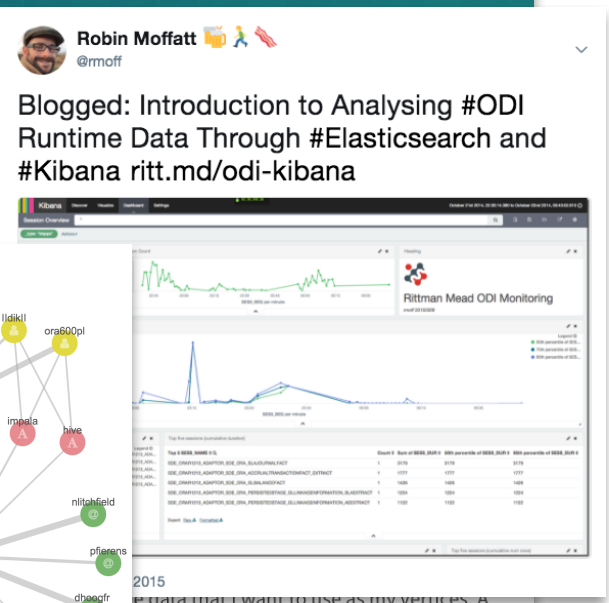
Introduction

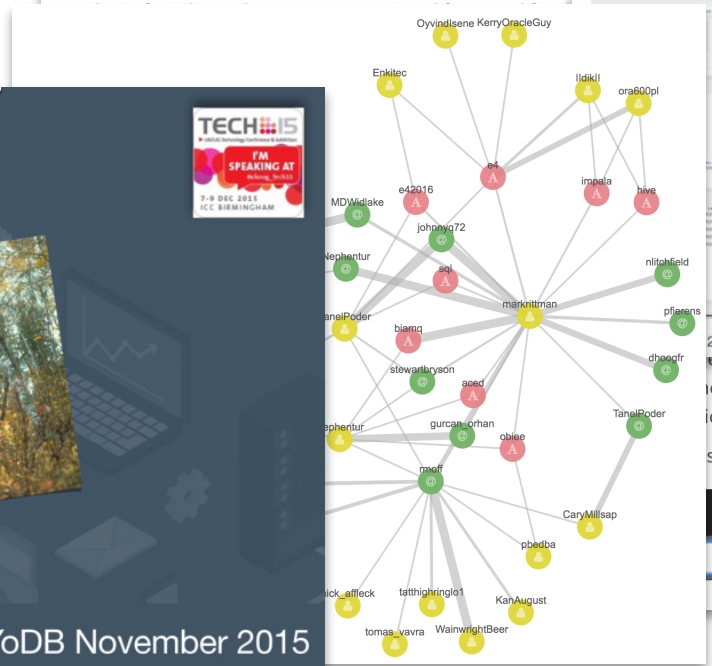
One of the great things about the Oracle data
available from it. Used in conjunction with go
on the database, it's a brilliant way – arguably
performance problems that arise. By analysir
more about the load that it is under, and by l
opportunities for it.

Experiments with Elastic's Graph Tool

29 APRIL 2016 on Technical, Rittman Mead Life, Business Insights, elasticsearch, kibana, twitter, graph

Elastic announced their Graph tool at ElastiCON 2016 (se
bundles Graph along with other helper tools such as Shie
Elasticsearch's capabilities, enabling the user to **explore**


Robin Moffatt
@rmoff
Blogged: Introduction to Analysing #ODI Runtime Data Through #Elasticsearch and #Kibana ritt.md/odi-kibana
Rittman Mead ODI Monitoring



Data Discovery & Systems Diagnostics with the ELK Stack

TECHIS I'M SPEAKING AT 7-9 DEC 2015 ICC BIRMINGHAM

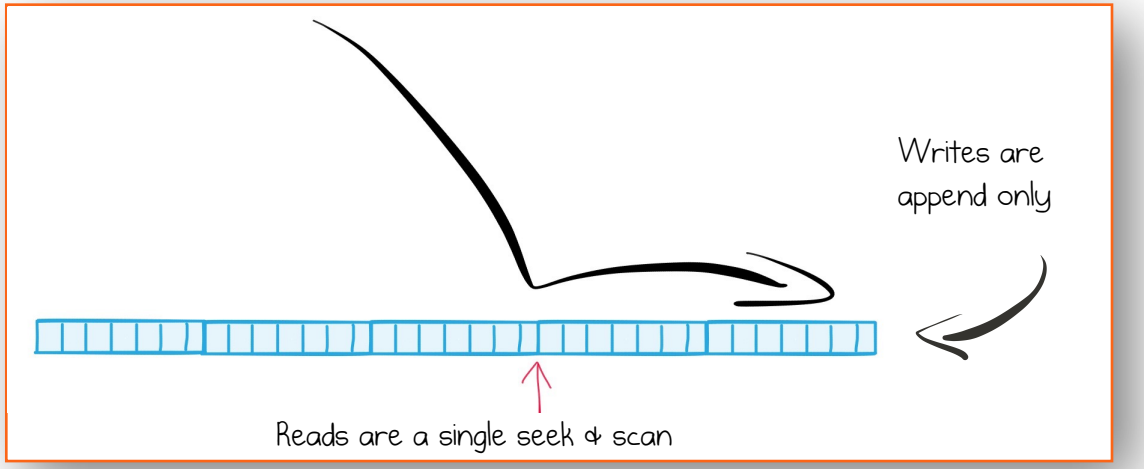
Robin Moffatt, Principal Consultant, Rittman Mead | YoDB November 2015

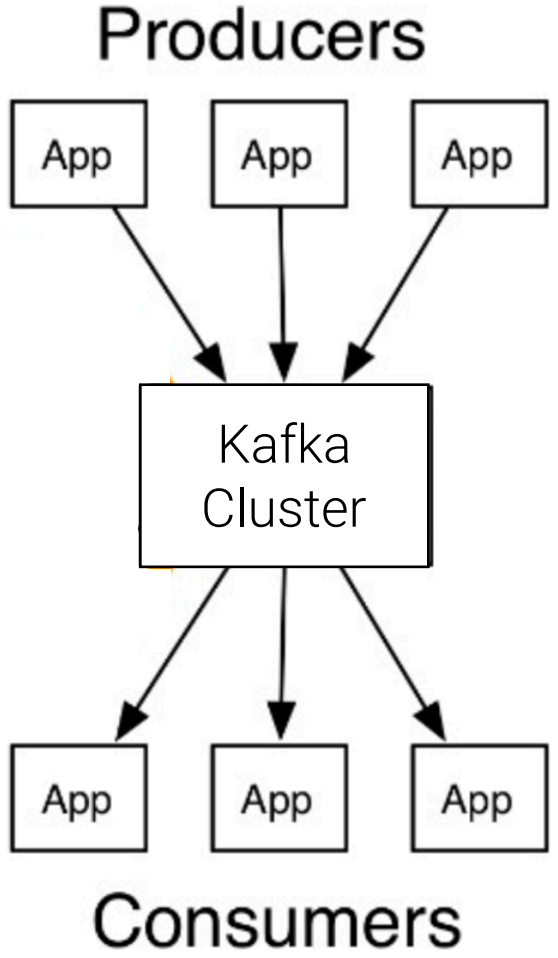
info@rittmanmead.com www.rittmanmead.com @rittmanmead rittmanmead

confluent

Kafka

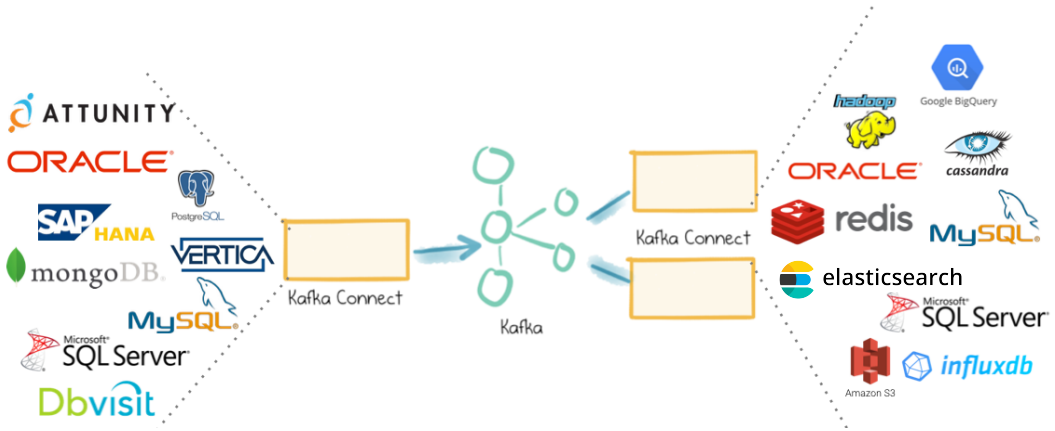
A **Distributed Commit Log**. Publish and subscribe to streams of records. Highly scalable, high throughput. Supports transactions. Persisted data.





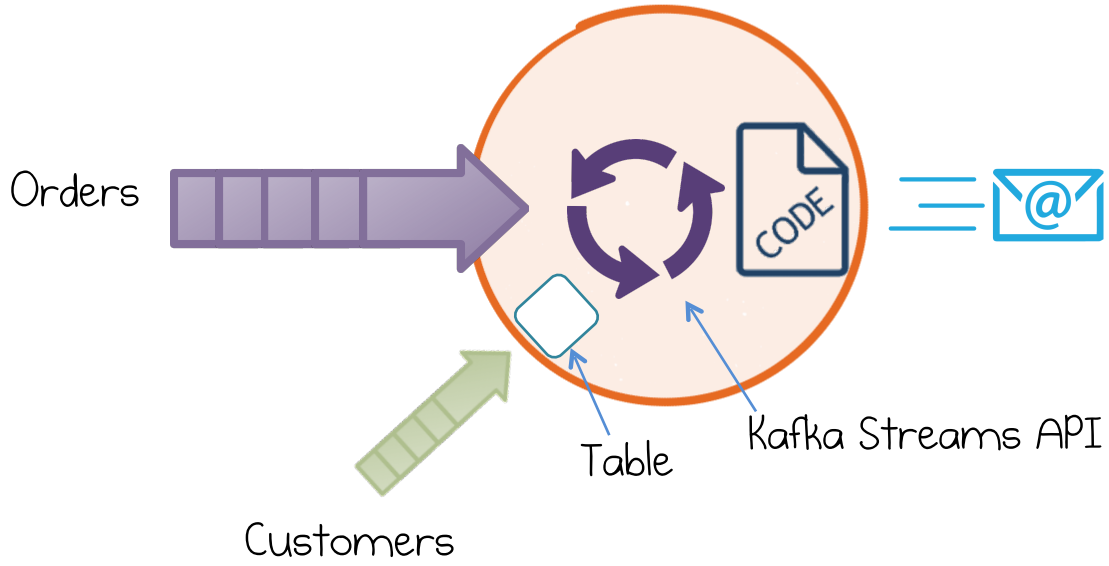
Kafka Connect API

Reliable and scalable integration of Kafka with other systems – no coding required.

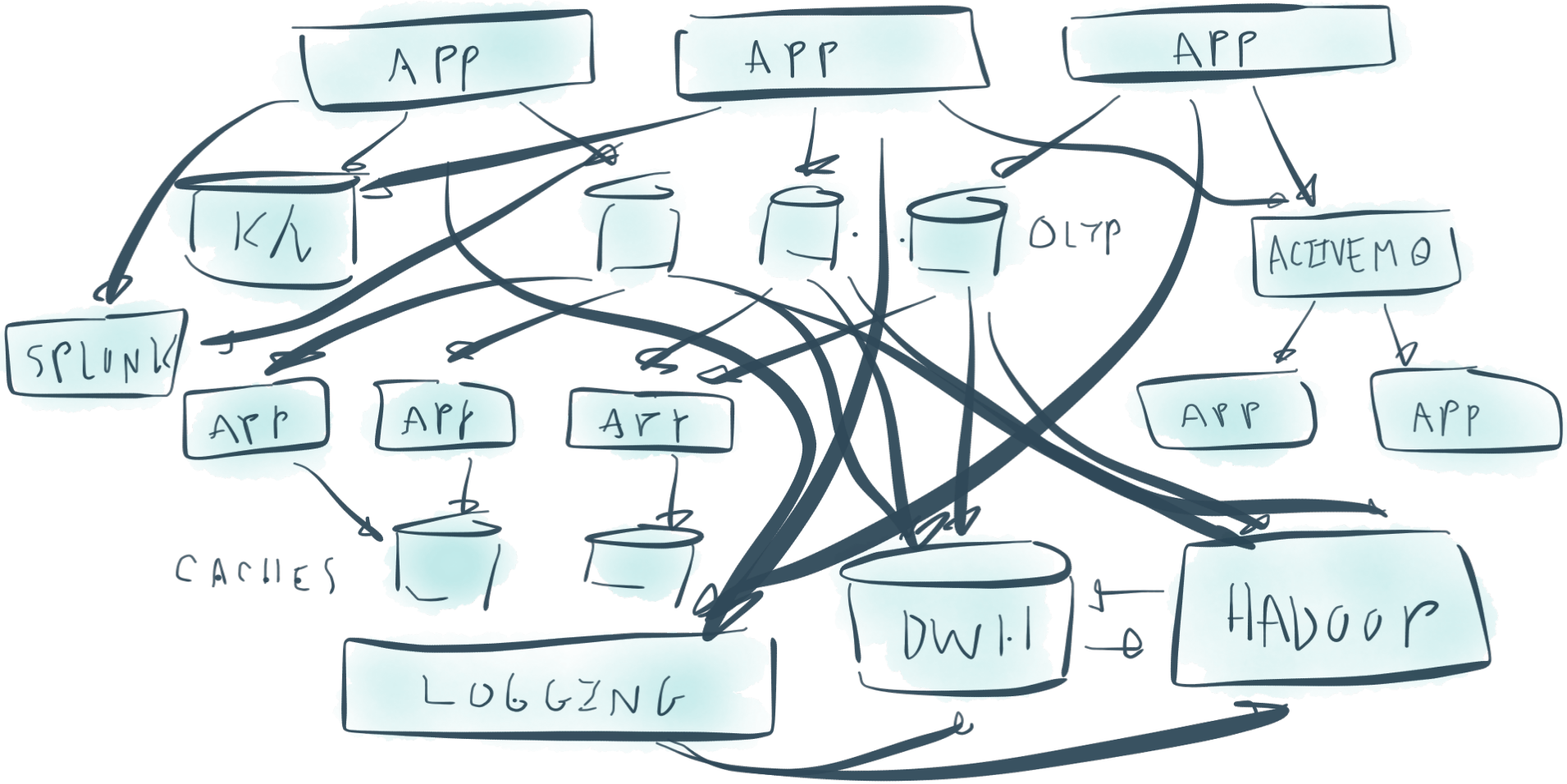


Kafka Streams API

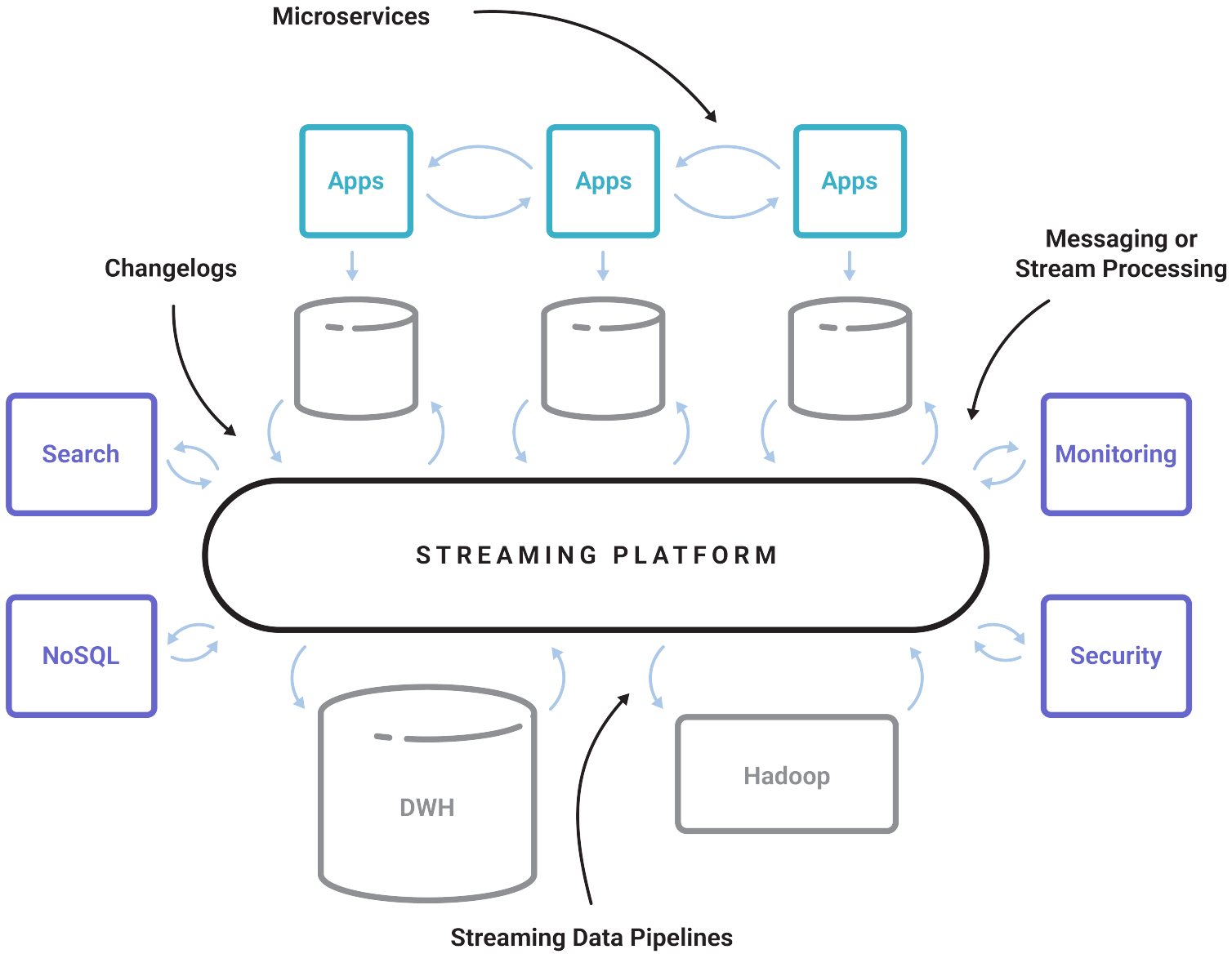
Write standard Java applications & microservices to process your data in real-time



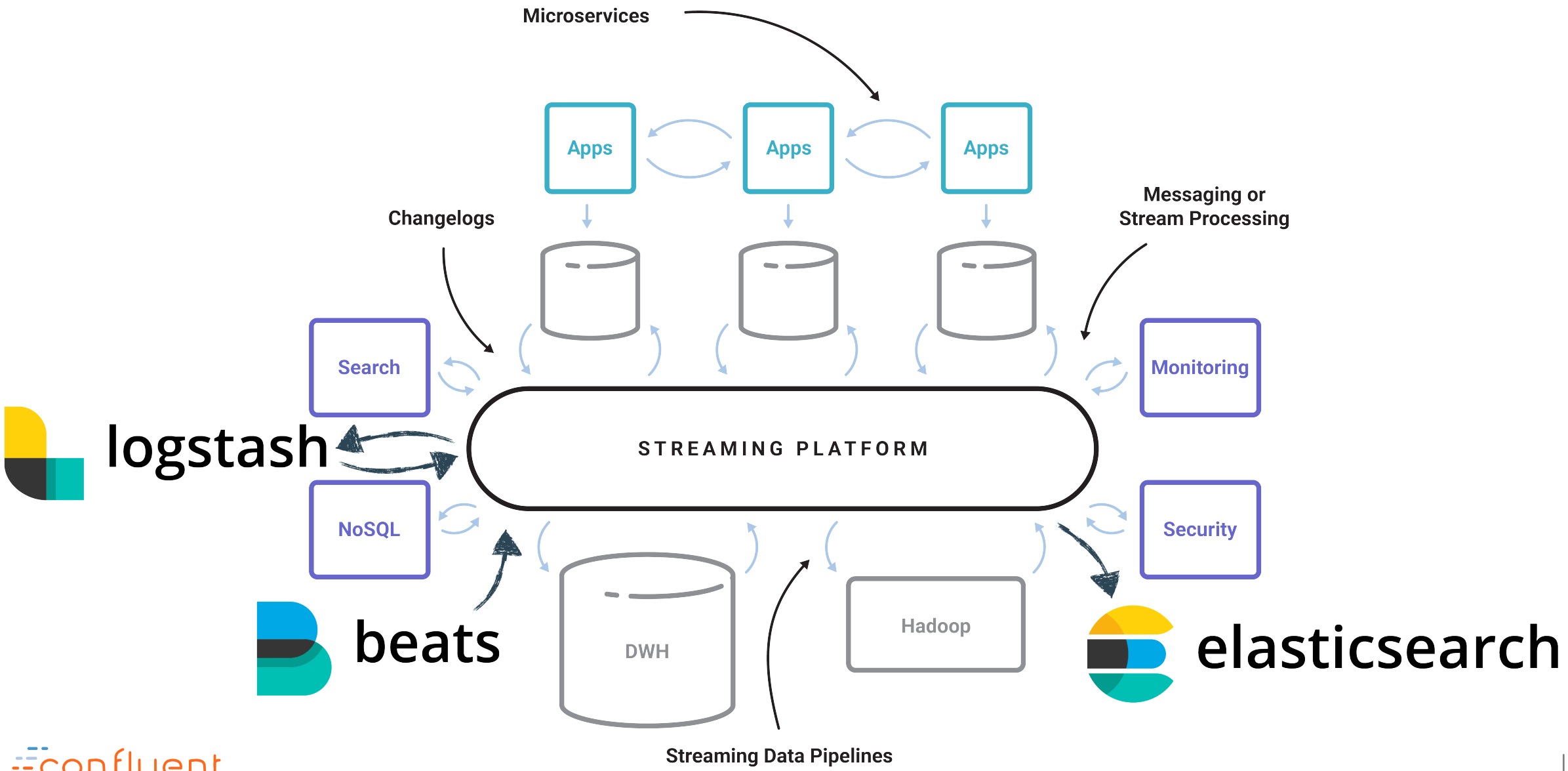
Many Systems are a bit of a mess…



The Streaming Platform

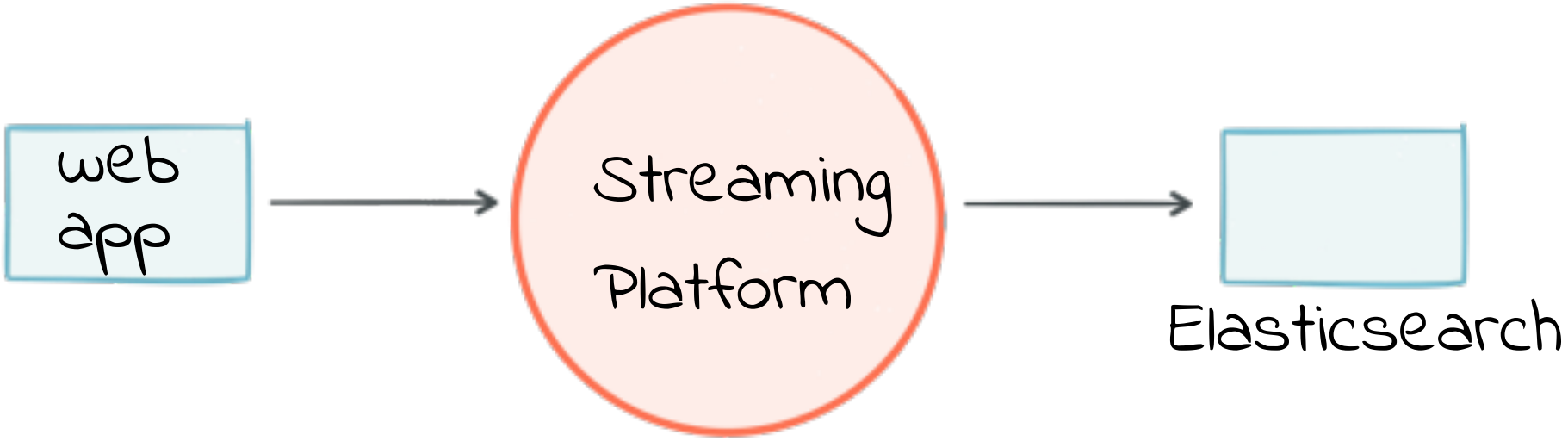


The Streaming Platform



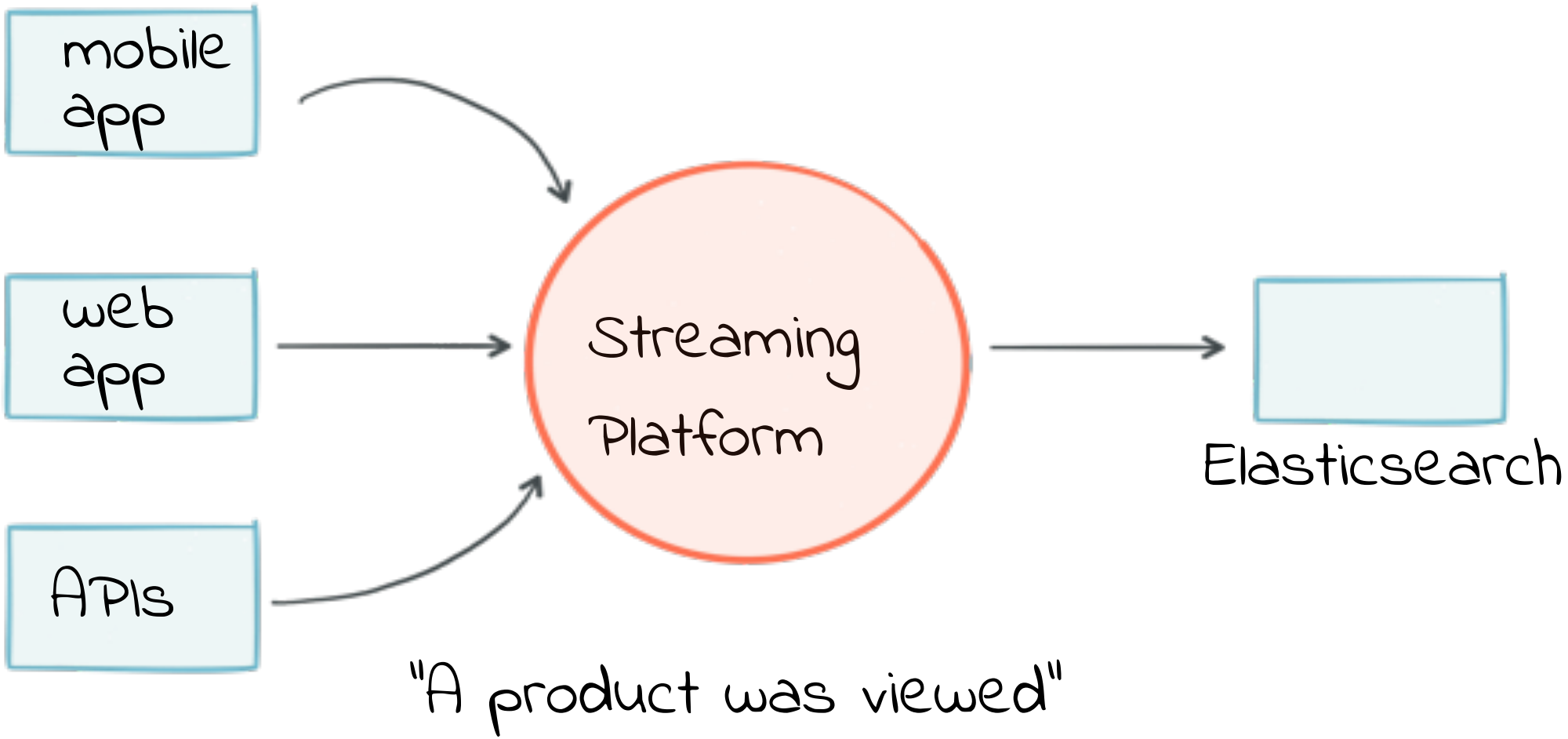
Why Kafka & Elastic?

Event-Centric Thinking

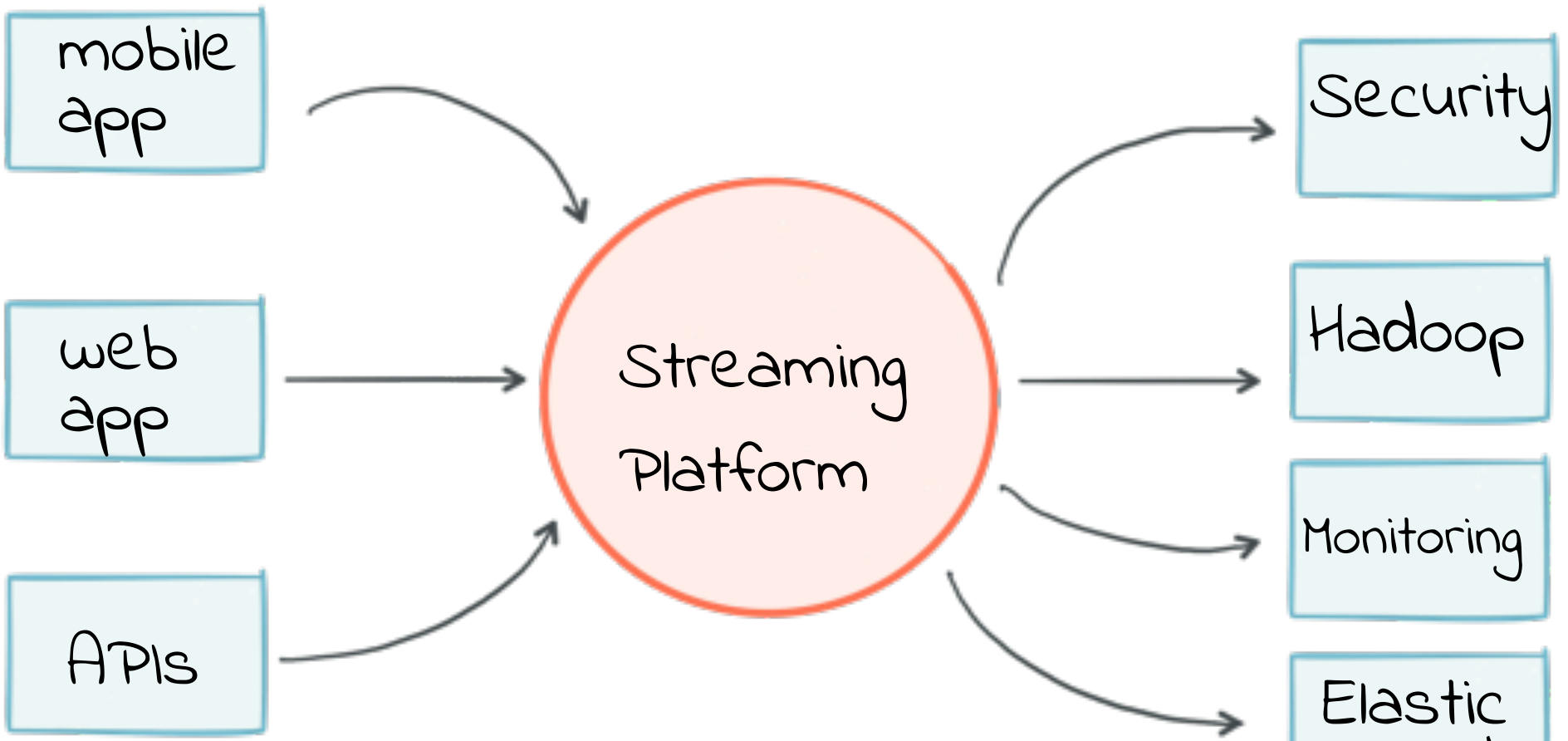


"A product was viewed"

Event-Centric Thinking



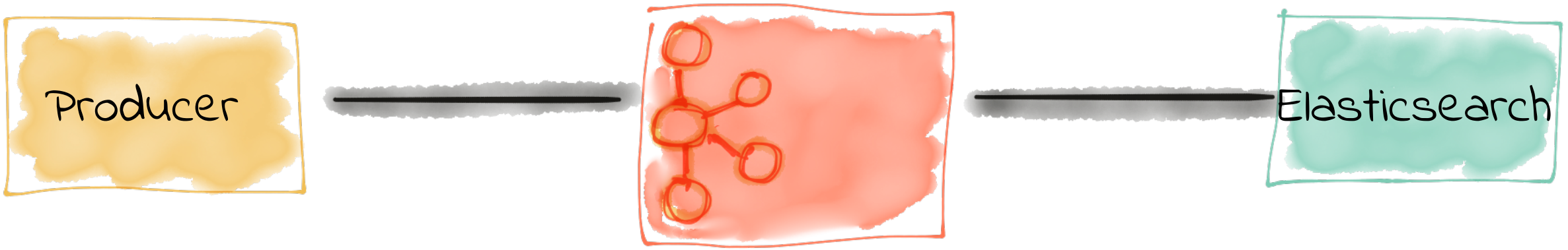
Event-Centric Thinking



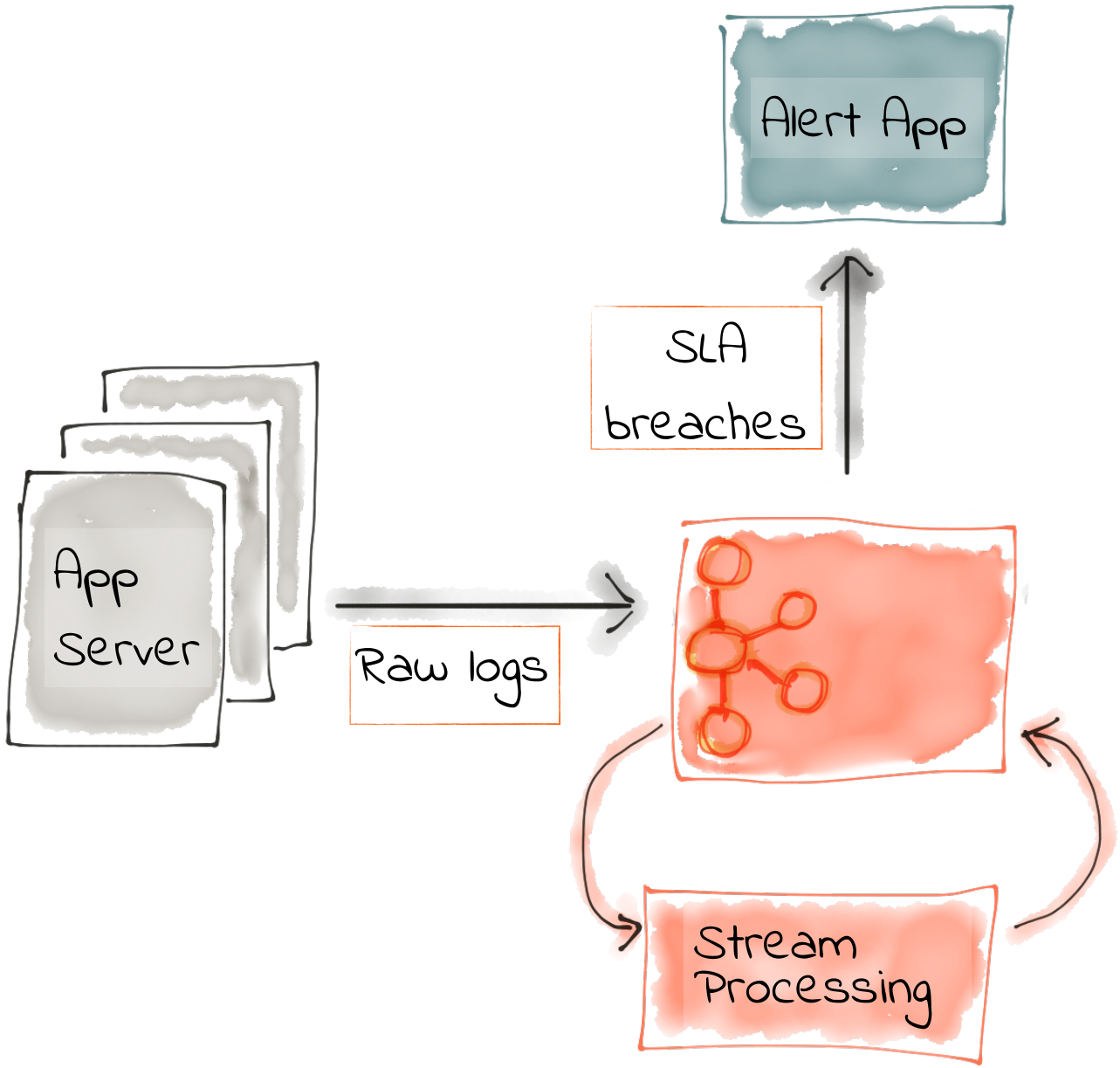
"A product was viewed"

System Availability and Event Buffering

System Availability and Event Buffering

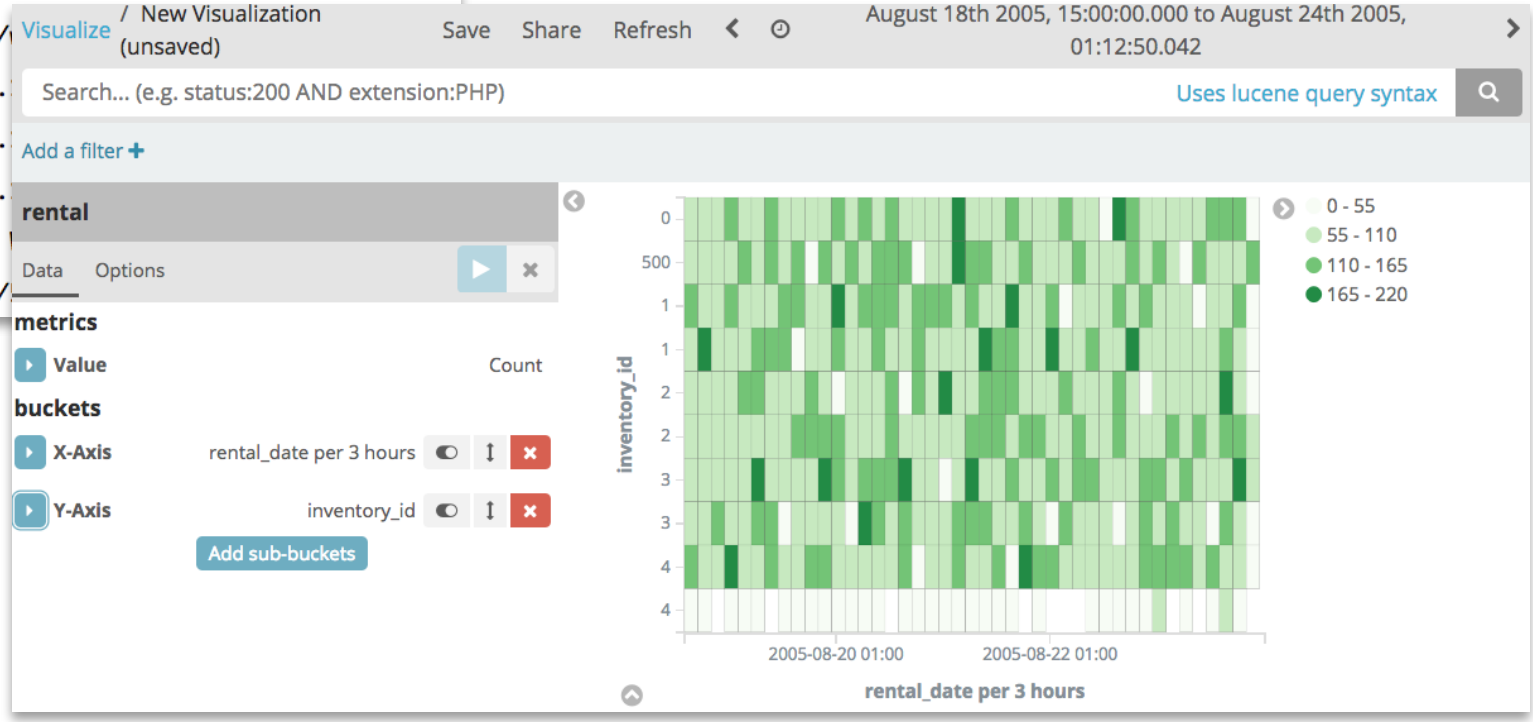


Native Stream Processing



Visualise & Analyse data from Kafka


```
{"REQUEST":"/index.rdf/","AGENT":"\"Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHT
{"REQUEST":"/index.php/?format=feed&type=rss","AGENT":"\"Mozilla/5.0 (Windows NT 6.3; WOW64) Ap
{"REQUEST":"/index.php/feed/","AGENT":"\"Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36
{"REQUEST":"/wp-login.php/","AGENT":"\"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100
{"REQUEST":"/wp-login.php/","AGENT":"\"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100
{"REQUEST":"/apple-touch-icon.png/","AGENT":"\"MobileSafari/604.1 CFNetwork/894 Darwin/17.4.0\"
{"REQUEST":"/apple-touch-icon.png/","AGENT":"\"Safari/12604.5.6.1.1 CFNetwork/811.8 Darwin/16.7
{"REQUEST":"/w00tw00t.at.ISC.SANS.DFind:)","AGENT":"\"-\"","RESPONSE":400,"TIMESTAMP":"24/Jan/2
{"REQUEST":"/robots.txt","AGENT":"\"Sogou web spider/4.0(+http://
{"REQUEST":"/spa112.cfg/","AGENT":"\"PolycomVVX-VVX_500-UA/5.2.2.
{"REQUEST":"/wp-login.php/","AGENT":"\"Mozilla/5.0 (Windows NT 6.
{"REQUEST":"/wp-login.php/","AGENT":"\"Mozilla/5.0 (Windows NT 6.
{"REQUEST":"/index.rdf/","AGENT":"\"Mozilla/5.0 (Windows NT 6.3;
{"REQUEST":"/index.php/?format=feed&type=rss","AGENT":"\"Mozilla/
```



confluent

Integrating Elastic and Kafka

Integrating Elastic with Kafka - Beats, Logstash

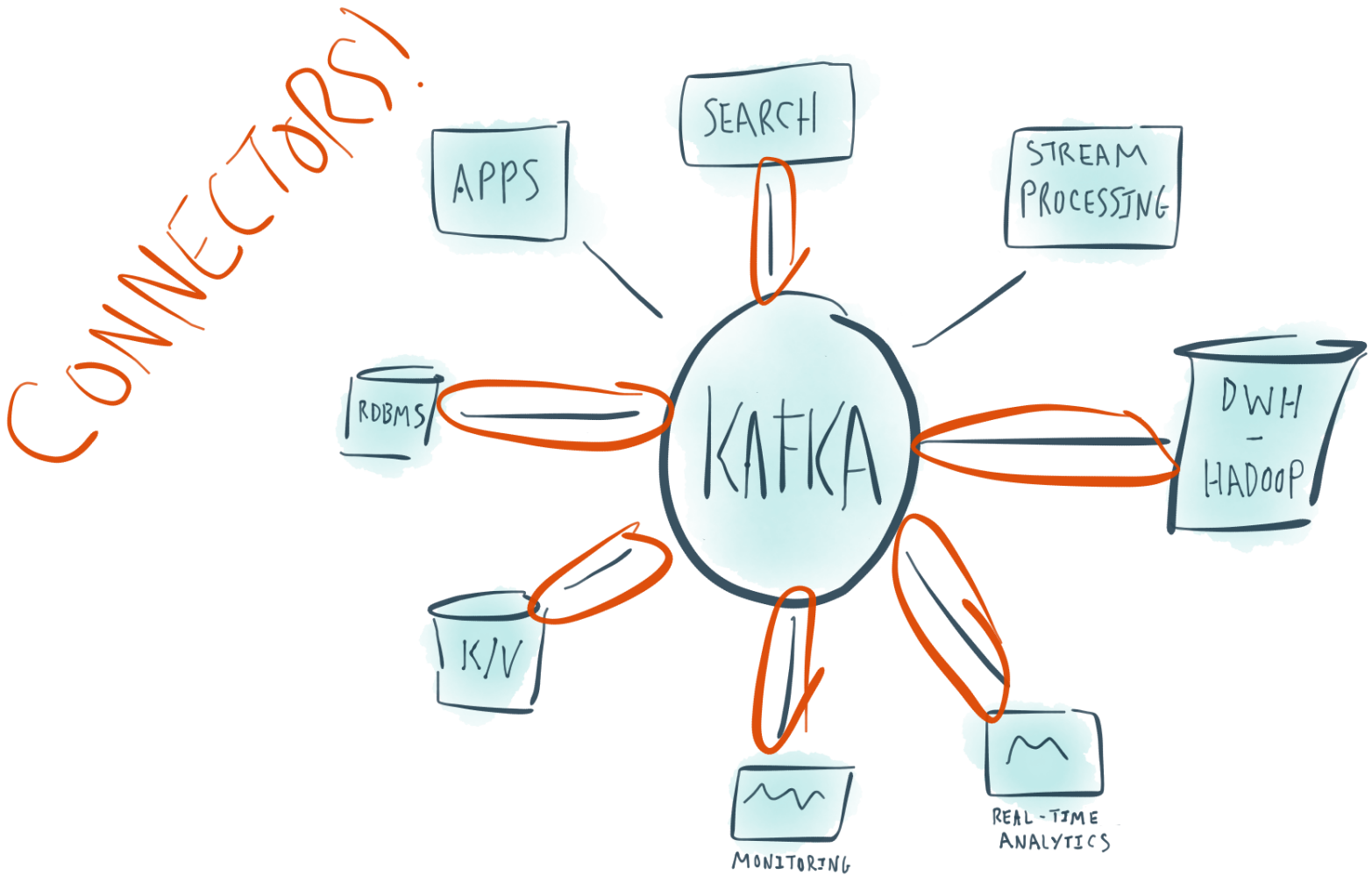
Beats

```
output.kafka:
  hosts: ["localhost:9092"]
  topic: 'logs'
  required_acks: 1
```

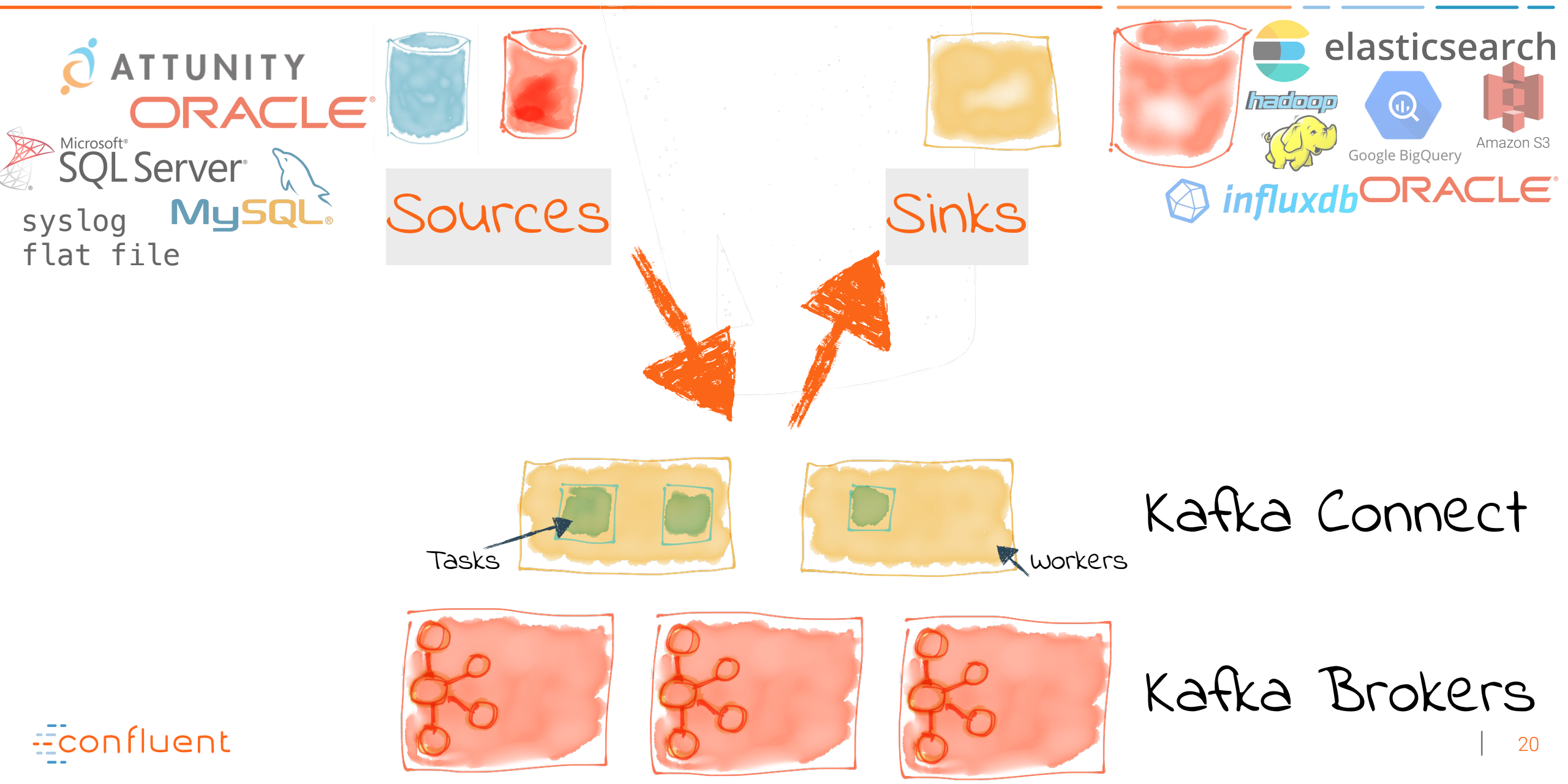
Logstash

```
output {
 kafka {
    topic_id => "logstash_logs_json"
    bootstrap_servers => "localhost:9092"
    codec => json
  }
}
```

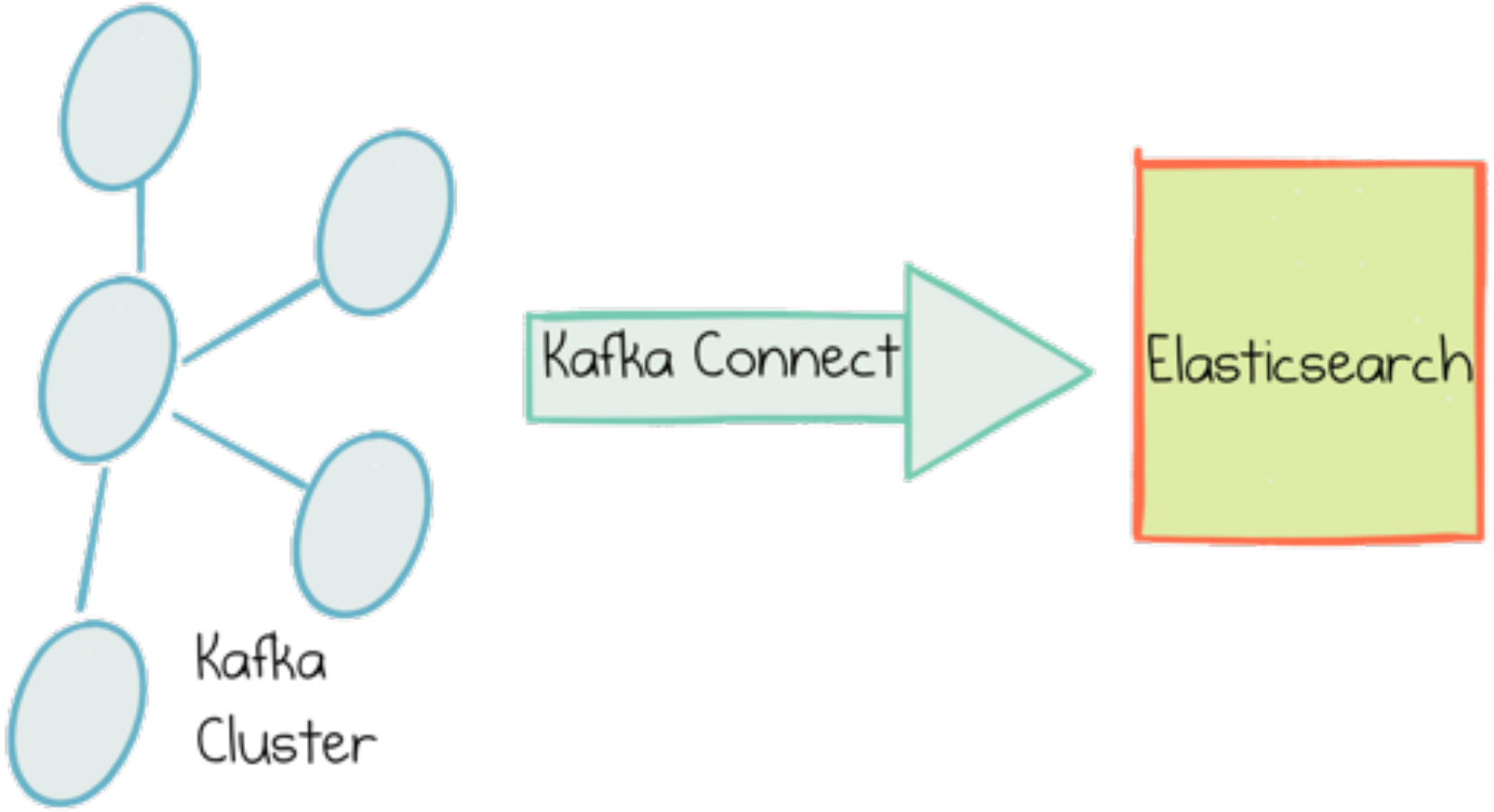
STREAMING PLATFORM



Kafka Connect



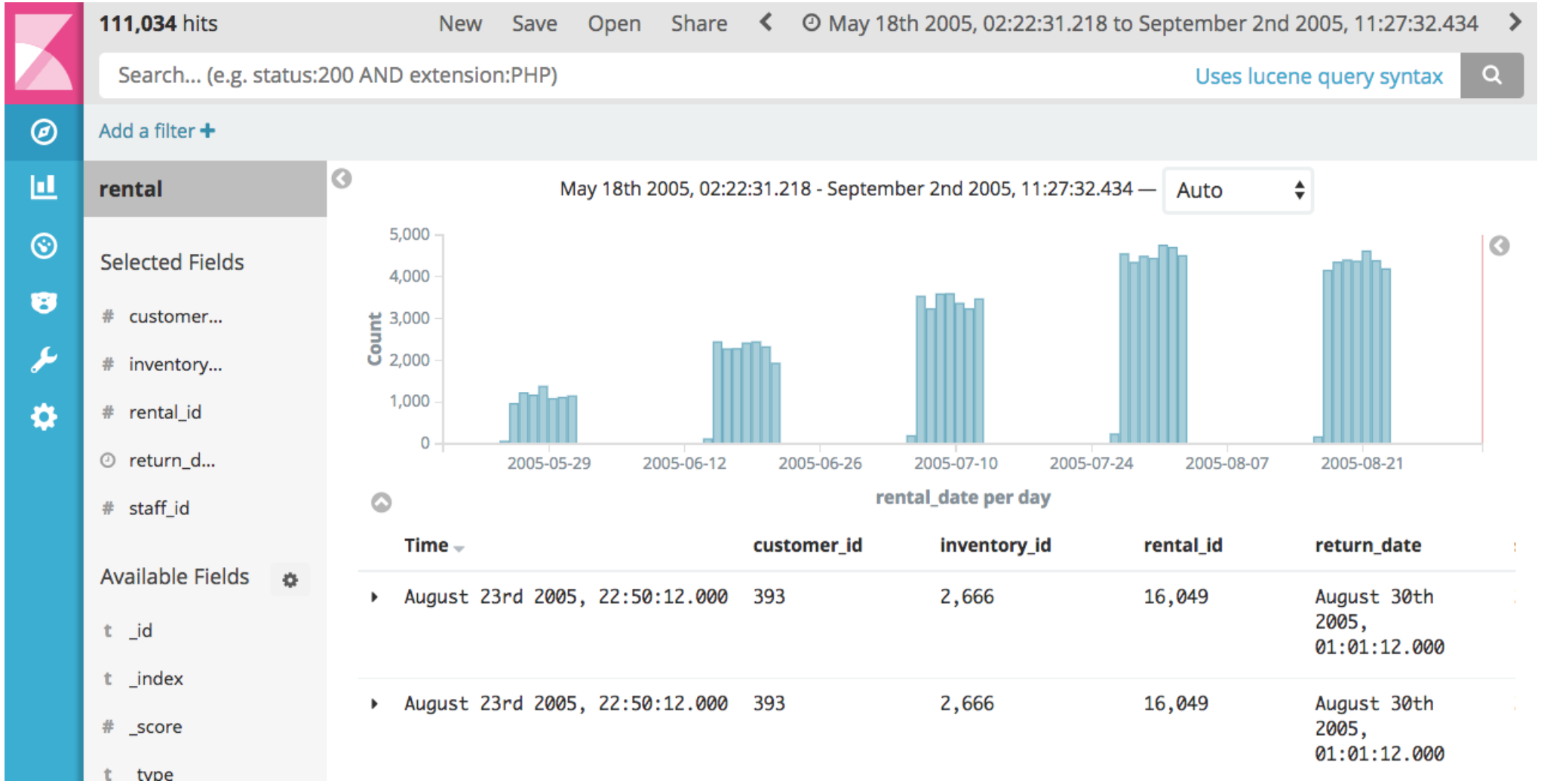
Kafka -> Elasticsearch



Kafka Connect's Elasticsearch Sink

```
{
  "name": "es-sink",
  "config": {
        "connector.class":
         "io.confluent.connect.elasticsearch.ElasticsearchSinkConnector",
        "connection.url": "http://localhost:9200",
        "type.name": "type.name=kafka-connect",
        "topics": "foobar"
        }
}
```

Kafka Connect to stream Kafka Topics to Elasticsearch



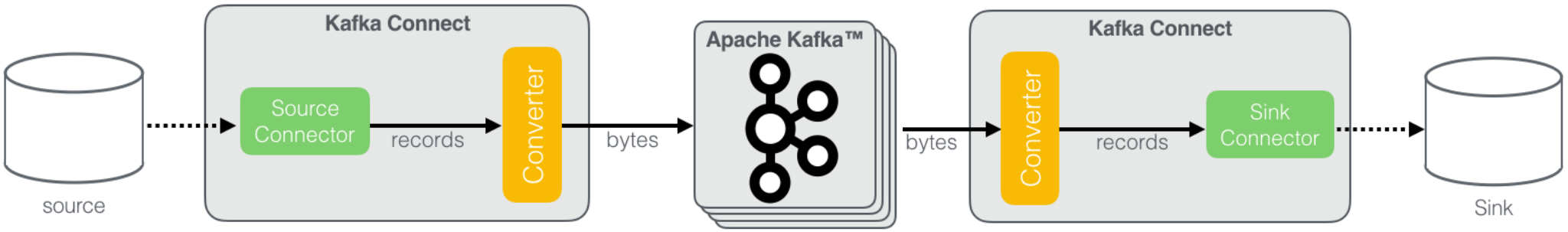
Kafka Connect

Elasticsearch Sink Properties

https://docs.confluent.io/current/connect/connect-elasticsearch/docs/configuration_options.html

Sink properties : Converters

- Kafka Connect uses **pluggable converters** for both message key and value deserialisation
  - Json, Avro, String, Protobuf, etc

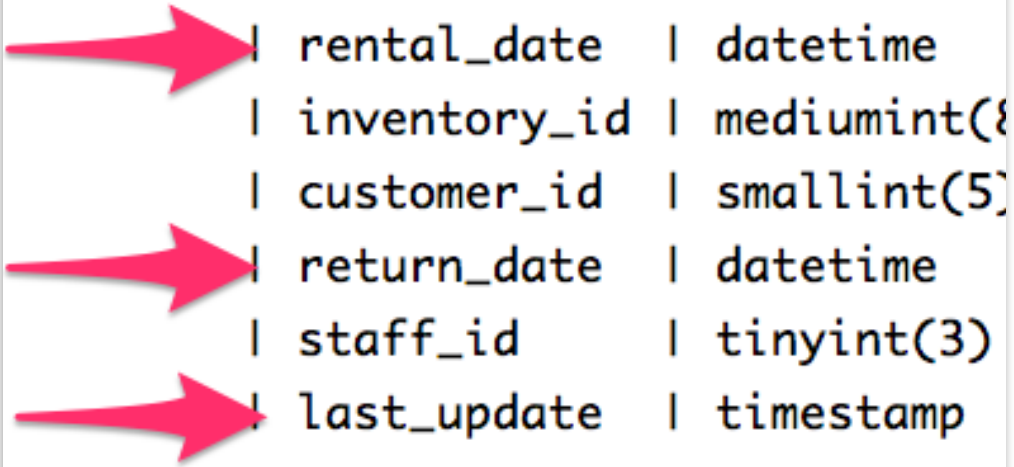


- Specify the converter in the Kafka Connect configuration, e.g.

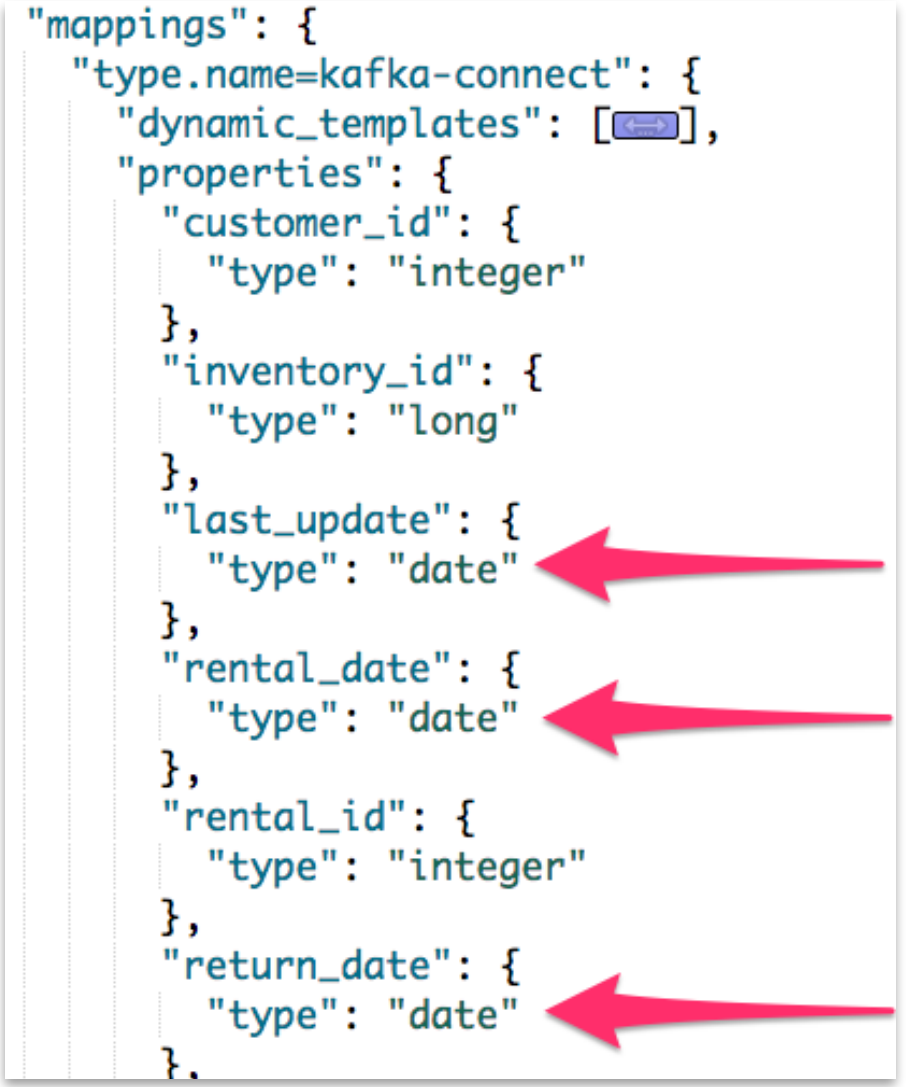
```
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
```

Schemas & Document Mappings

```
mysql> describe rental;
+--------------+-----------
| Field        | Type
+--------------+-----------
| rental_id    | int(11)
| rental_date  | datetime
| inventory_id | mediumint(
| customer_id  | smallint(5
| return_date  | datetime
| staff_id     | tinyint(3)
| last_update  | timestamp
+--------------+-----------
```



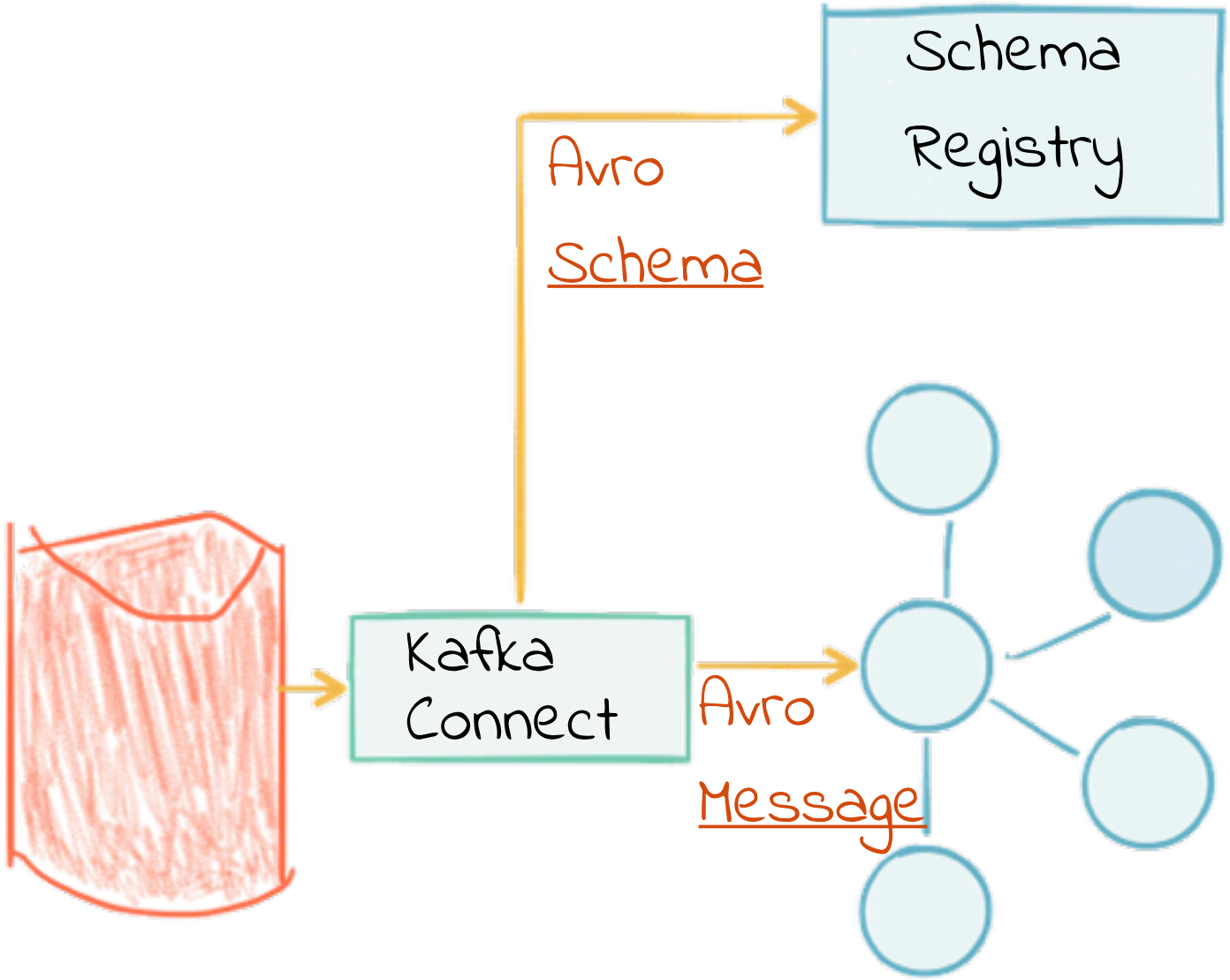
```
"mappings": {
  "type.name=kafka-connect": {
    "dynamic_templates": [⟺],
    "properties": {
      "customer_id": {
        "type": "integer"
      },
      "inventory_id": {
        "type": "long"
      },
      "last_update": {
        "type": "date"
      },
      "rental_date": {
        "type": "date"
      },
      "rental_id": {
        "type": "integer"
      },
      "return_date": {
        "type": "date"
      },
```



Schemas in Kafka Connect - JSON

```
{"schema":
  {"type":"struct",
  "fields":[{"type":"int32","optional":true,"field":"c1"},
            {"type":"string","optional":true,"field":"c2"},
            {"type":"int64","optional":false,
              "name":"org.apache.kafka.connect.data.Timestamp","field":"create_ts"},
            {"type":"int64","optional":false,
              "name":"org.apache.kafka.connect.data.Timestamp","field":"update_ts"}],
  "optional":false,
  "name":"foobar"
  },
"payload":{ "c1":100,
            "c2":"bar",
            "create_ts":1516747629000,
            "update_ts":1516747629000}
}
```

Kafka Connect + Schema Registry = WIN



Schemas in Kafka Connect - Avro & Confluent Schema Registry


```
{"subject":"mysql-foobar-value","version":1,"id":141,"schema":"{\"type\":\"record\",
\"name\":\"foobar\",\"fields\":[{\"name\":\"c1\",\"type\":[\"null
[\"null\",\"string\"],\"default\":null},{\"name\":\"create_ts\",\"type\":{\"type\":
\"long\",\"connect.version\":1,\"connect.name\":\"org.apache.kafka
llis\"}},{\"name\":\"update_ts\",\"type\":{\"type\":\"long\",\"connect.version\":
1,\"connect.name\":\"org.apache.kafka.connect.data.Timestamp\",\"log
obar\"}"}
```

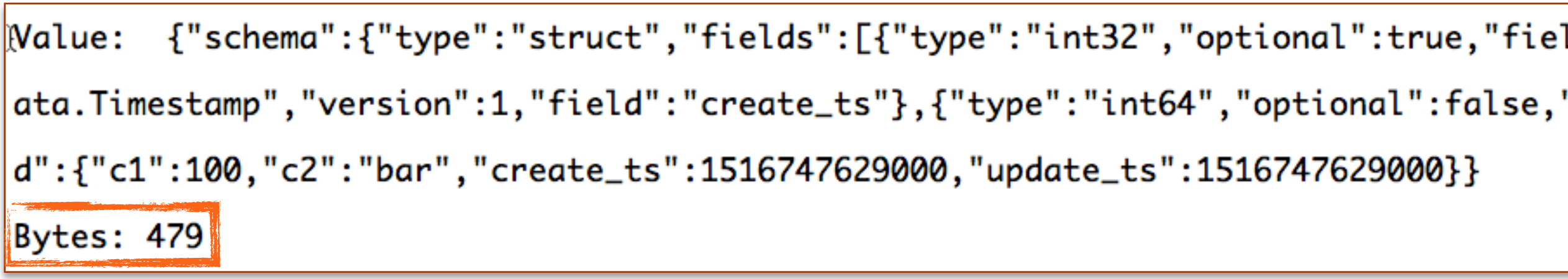

Schema

Message

Avro & JSON schema handling

Avro

JSON

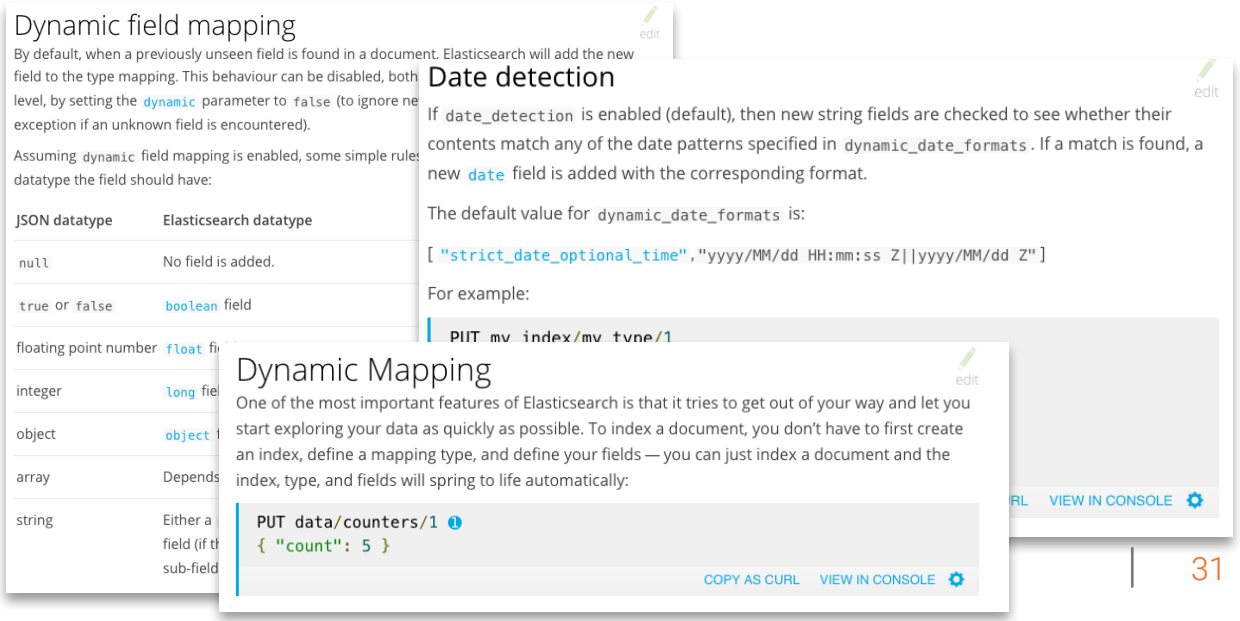


Sink properties : Schema/Mapping handling

`schema.ignore=true`

- **Kafka Connect will let Elasticsearch create the mapping**
- Elasticsearch uses dynamic mapping to guess datatypes
- Use dynamic templates to handle timestamps
  - Or explicitly create the document mapping beforehand

- Best used when source data is JSON (e.g. from logstash) without a compatible schema


Dynamic field mapping
By default, when a previously unseen field is found in a document, Elasticsearch will add the new field to the type mapping. This behaviour can be disabled, both
level, by setting the dynamic parameter to false (to ignore ne
exception if an unknown field is encountered).
Assuming dynamic field mapping is enabled, some simple rules
datatype the field should have:
JSON datatype
Elasticsearch datatype
null
No field is added.
true or false
boolean field
floating point number
integer
object
array
string
Date detection
If date_detection is enabled (default), then new string fields are checked to see whether their contents match any of the date patterns specified in dynamic_date_formats. If a match is found, a new date field is added with the corresponding format.
The default value for dynamic_date_formats is:
["strict_date_optional_time","yyyy/MM/dd HH:mm:ss Z||yyyy/MM/dd Z"]
For example:
Dynamic Mapping
One of the most important features of Elasticsearch is that it tries to get out of your way and let you start exploring your data as quickly as possible. To index a document, you don't have to first create an index, define a mapping type, and define your fields — you can just index a document and the index, type, and fields will spring to life automatically:
PUT data/counters/1
{ "count": 5 }
COPY AS CURL
VIEW IN CONSOLE

Sink properties : Schema/Mapping handling

`schema.ignore=false`

- **Kafka Connect will create the document mapping using the schema provided**
- Therefore, you must provide a schema
  - Avro
  - JSON *with schema/payload format*
- Useful for preserving timestamps

- Use for end-to-end schema preservation when using Kafka Connect for ingest too (e.g. from RDBMS)

- Not currently supported with ES6

Sink properties : Document id

`key.ignore=false`

- Kafka Connect will use the message's key as the document id
- Therefore, your message must have a key
- Useful for storing latest version of a record only
  - e.g. account balance

`key.ignore=true`

- Kafka Connect will specify the document id based as a tuple of topic/partition/offset
- Useful if you don't have message keys

An order...

ID	Product	Shipping Address	Status
42	iPad	--	New
42	iPad	**29 Acacia Road**	**Packing**
42	iPad	29 Acacia Road	**Shipped**
42	iPad	29 Acacia Road	**Delivered**

Store every state change

`key.ignore=true`

_id	ID	Product	Shipping Address	Status
01	42	iPad	--	New
02	42	iPad	**29 Acacia Road**	**Packing**
03	42	iPad	29 Acacia Road	**Shipped**
04	42	iPad	29 Acacia Road	**Delivered**

Update document in place

`key.ignore=false`

_id	ID	Product	Shipping Address	Status
42	42	iPad	--	New

Update document in place

`key.ignore=false`

_id	ID	Product	Shipping Address	Status
42	42	iPad	**29 Acacia Road**	**Packing**

Update document in place

`key.ignore=false`

_id	ID	Product	Shipping Address	Status
42	42	iPad	29 Acacia Road	**Shipped**

Update document in place

`key.ignore=false`

_id	ID	Product	Shipping Address	Status
42	42	iPad	29 Acacia Road	**Delivered**

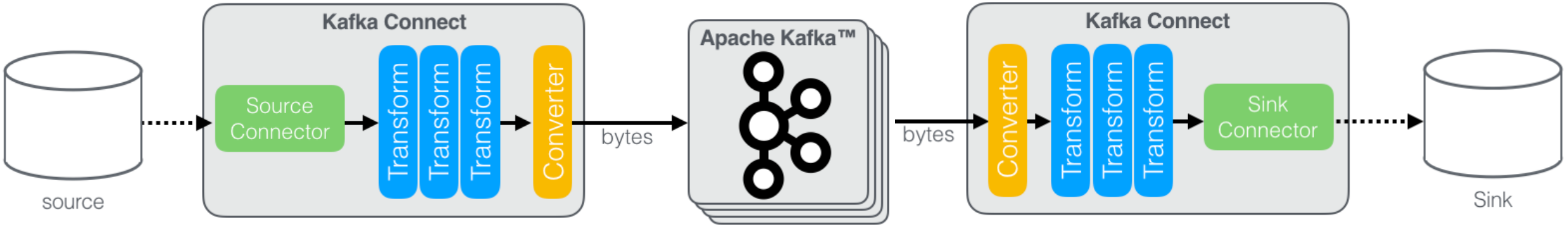
Sink properties : Index

```
topic.index.map=TOPIC:index,FOO:bar
```

- By default, Kafka Connect will use the topic name as the index
- Necessary to override if the topic is in capitals
- Useful to override for adherence with naming standards, etc

Single Message Transform (SMT) -- Extract, TRANSFORM, Load…

- Modify events *before storing* in Kafka:
  - Mask/drop sensitive information
  - Set partitioning key
  - Store lineage
  - Cast data types
- Modify events *going out* of Kafka:
  - Direct events to different Elasticsearch indexes
  - Mask/drop sensitive information
  - Cast data types to match destination



Customising target index name with Single Message Transforms

```
"transforms":"routeTS",
"transforms.routeTS.type":"org.apache.kafka.connect.transforms.TimestampRouter",
"transforms.routeTS.topic.format":"${topic}-${timestamp}",
"transforms.routeTS.timestamp.format":"YYYYMM"
```

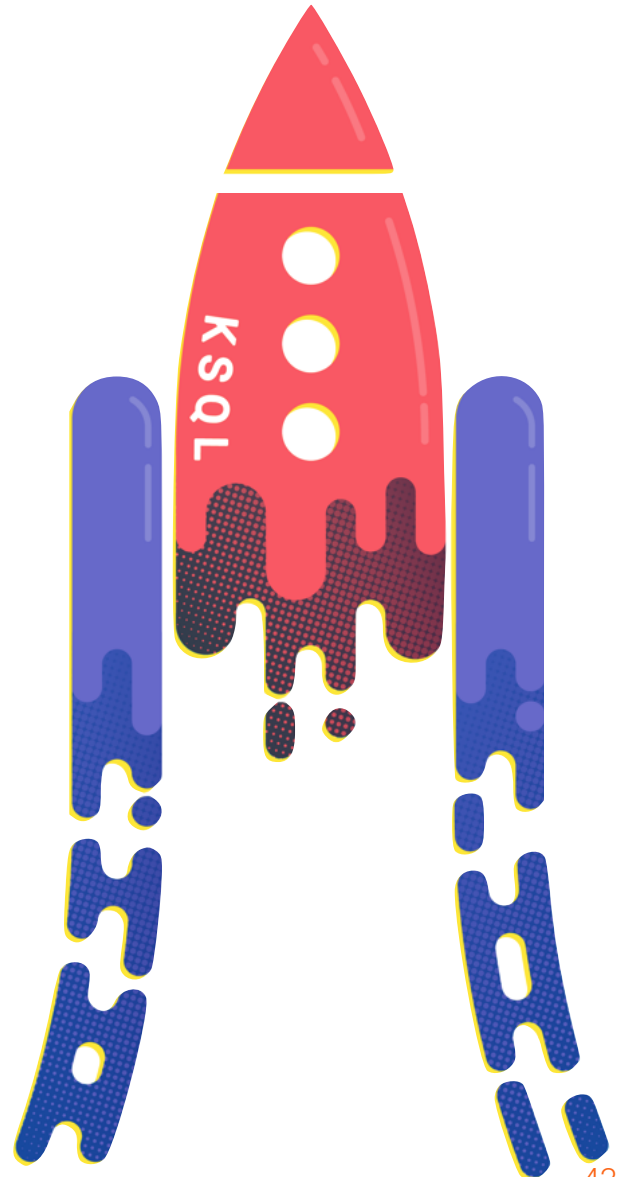
Source topic	Elasticsearch index
sales	sales-201801
sales	sales-201802

```
"transforms": "dropPrefix",
"transforms.dropPrefix.type":"org.apache.kafka.connect.transforms.RegexRouter",
"transforms.dropPrefix.regex":"DC?-(.*)-avro",
"transforms.dropPrefix.replacement":"$1"
```

Source topic	Elasticsearch index
DC1-sales-avro	sales
DC2-sales-avro	sales

KSQL: a Streaming SQL Engine for Apache Kafka® from Confluent

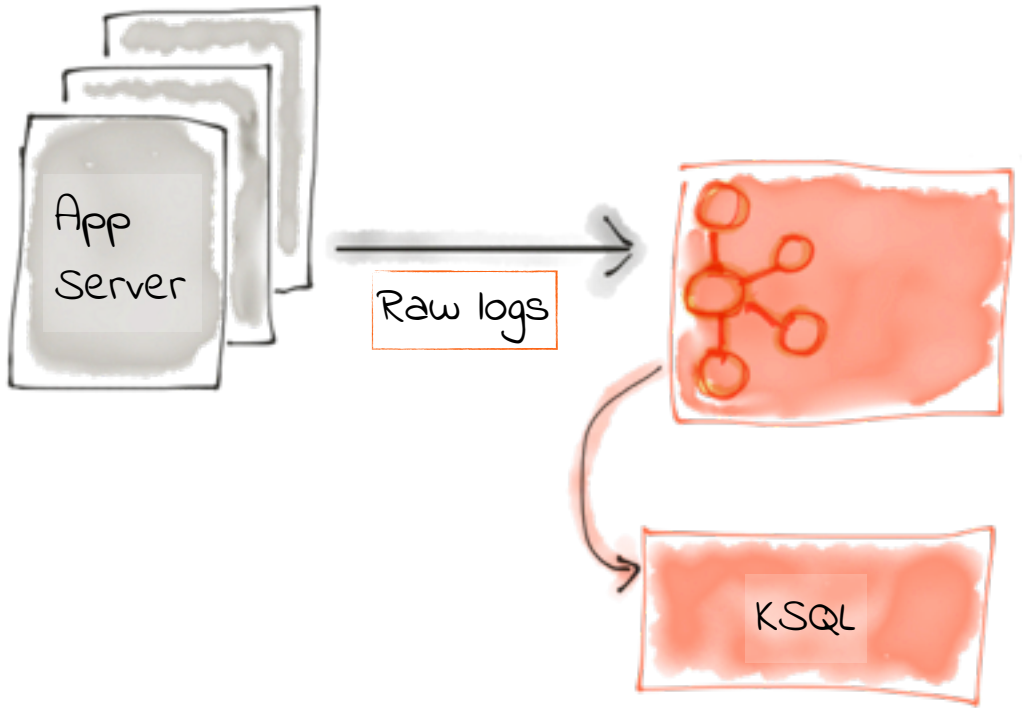
- Enables stream processing with zero coding required
- The simplest way to process streams of data in real-time
- Powered by Kafka: scalable, distributed, battle-tested
- All you need is Kafka–No complex deployments of bespoke systems for stream processing



KSQL: the Simplest Way to Do Stream Processing

```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 SECONDS)
GROUP BY card_number
HAVING count(*) > 3;
```

Streaming Transformations with KSQL



KSQL for querying and transforming log files

```
ksql> CREATE STREAM LOGS
      (REQUEST VARCHAR, AGENT VARCHAR, RESPONSE INT,
      TIMESTAMP VARCHAR)
      WITH (KAFKA_TOPIC='LOGSTASH_LOGS_JSON' ,
      VALUE_FORMAT='JSON');

 Message
----------------
 Stream created
----------------
```

KSQL for querying and transforming log files

```
ksql> SELECT REQUEST, RESPONSE
      FROM LOGS
      WHERE REQUEST LIKE '%jpg';

/content/images/2018/01/cow-and-calf.jpg | 200
/content/images/2016/02/IMG_4810-copy.jpg | 200
/content/images/2017/11/oggkaf01_sm.jpg | 200
/content/images/2016/06/IMG_7889-1.jpg | 200
/content/images/2016/02/IMG_4810-copy.jpg | 200
```

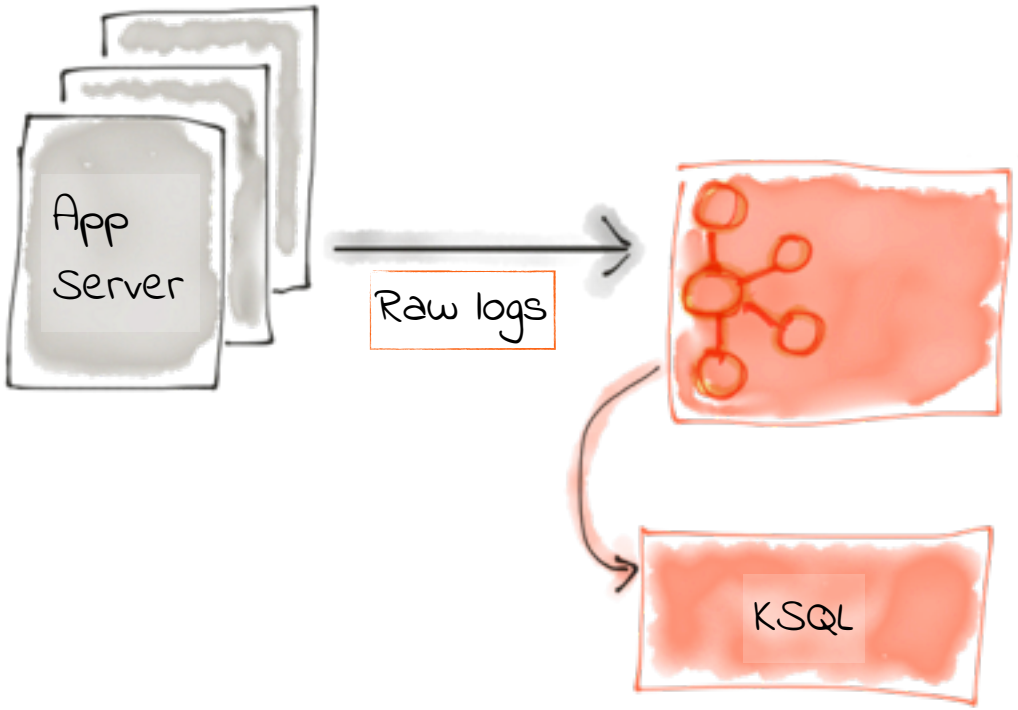
KSQL for querying and transforming log files

```
ksql> SELECT REQUEST, RESPONSE
      FROM LOGS
      WHERE RESPONSE > 400;

/2016/06/07/ | 404
/wp-login.php/ | 404
/spa112.cfg/ | 404
/spa122.cfg/ | 404
/never/gonna/give/you/up/ | 404
```

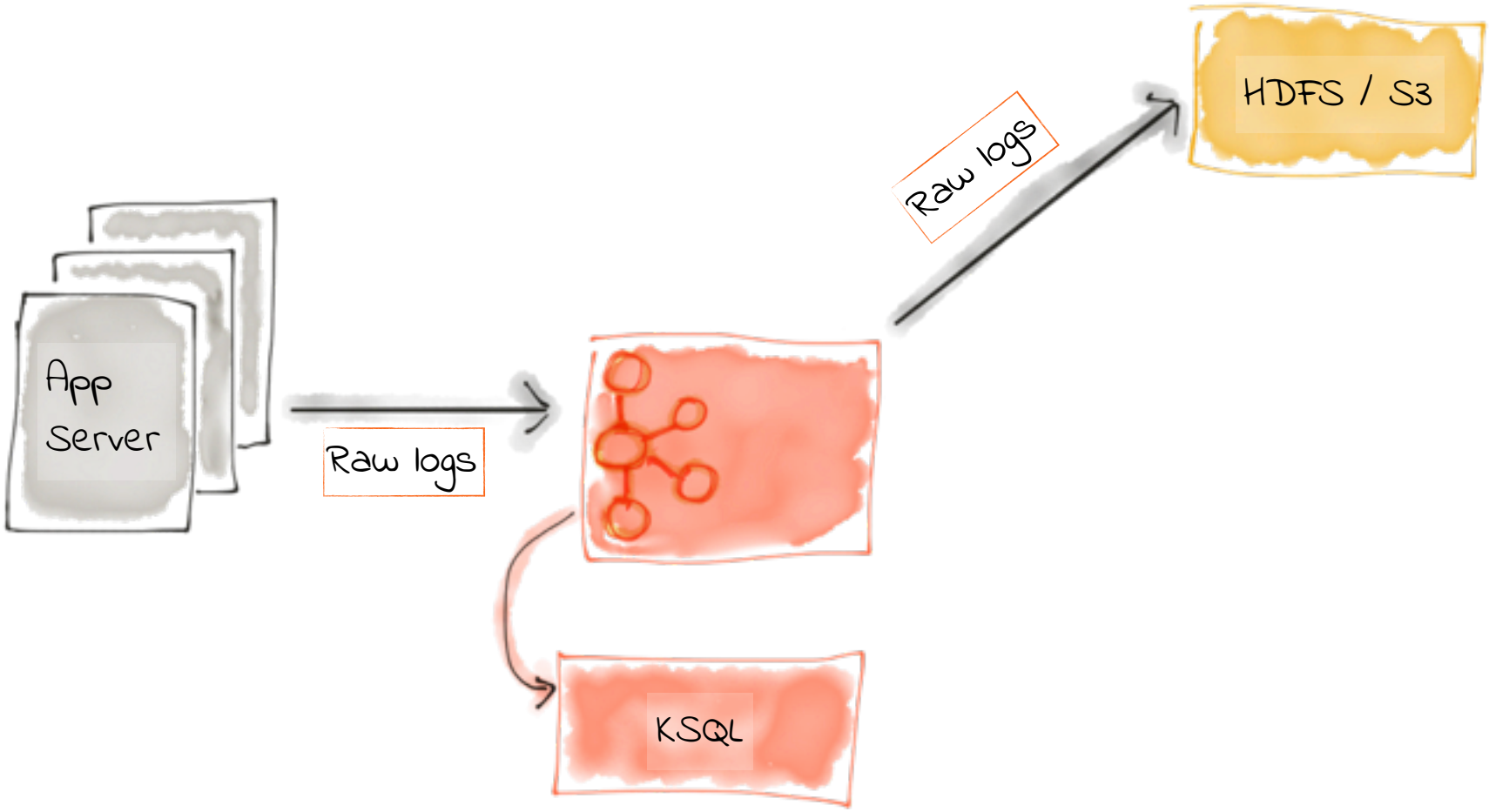
Creating streaming aggregates with KSQL

```
ksql> SELECT RESPONSE,COUNT(*) AS REQUEST_COUNT
      FROM LOGS
      WINDOW TUMBLING (SIZE 1 MINUTE)
      GROUP BY RESPONSE;

2018-01-23 19:00:00 | 304 | 20
2018-01-23 19:00:00 | 404 | 1
2018-01-23 19:01:00 | 304 | 9
2018-01-23 19:01:00 | 404 | 2
2018-01-23 19:01:00 | 418 | 1
```

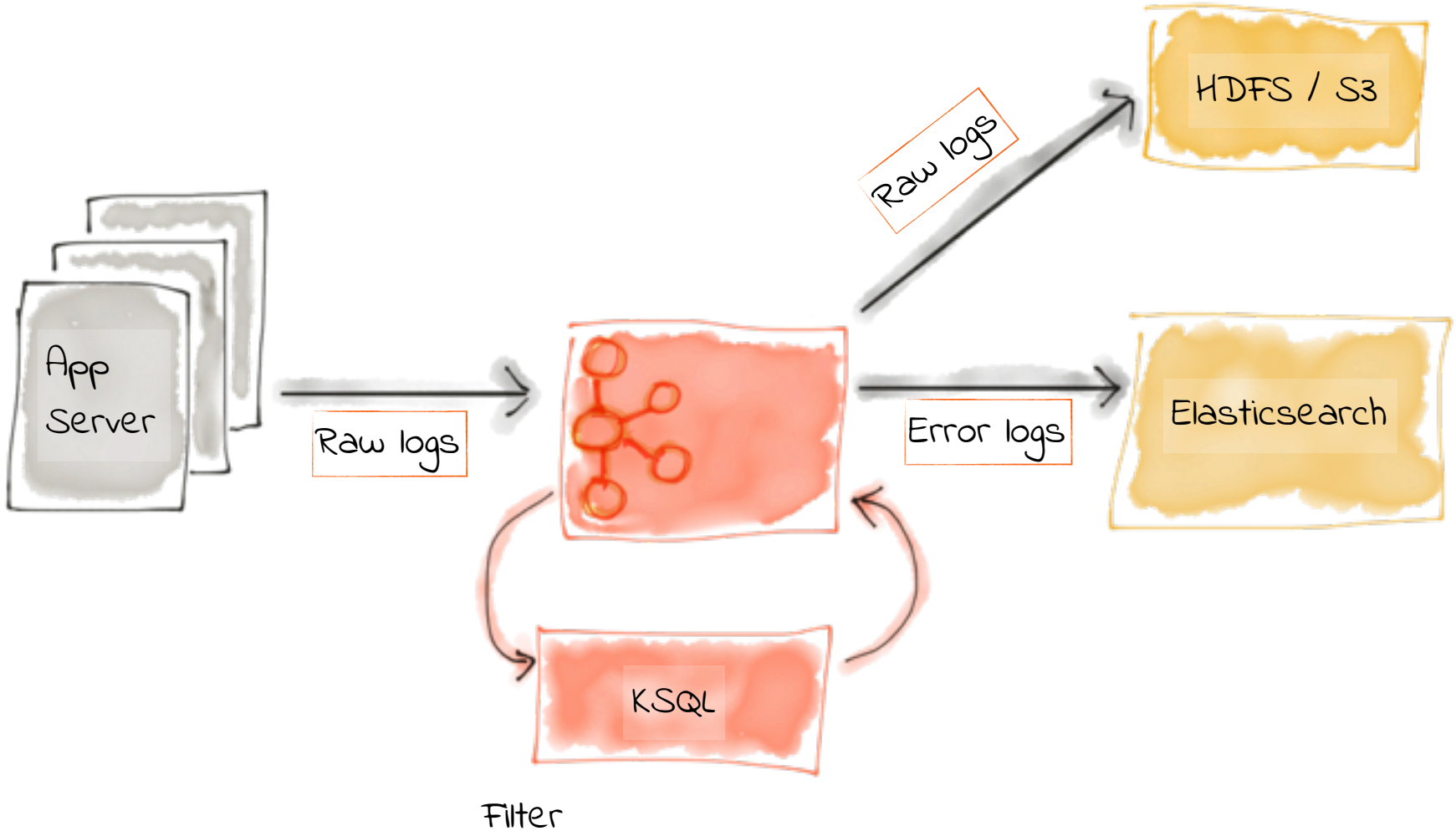
Streaming Transformations with KSQL



Streaming Transformations with KSQL



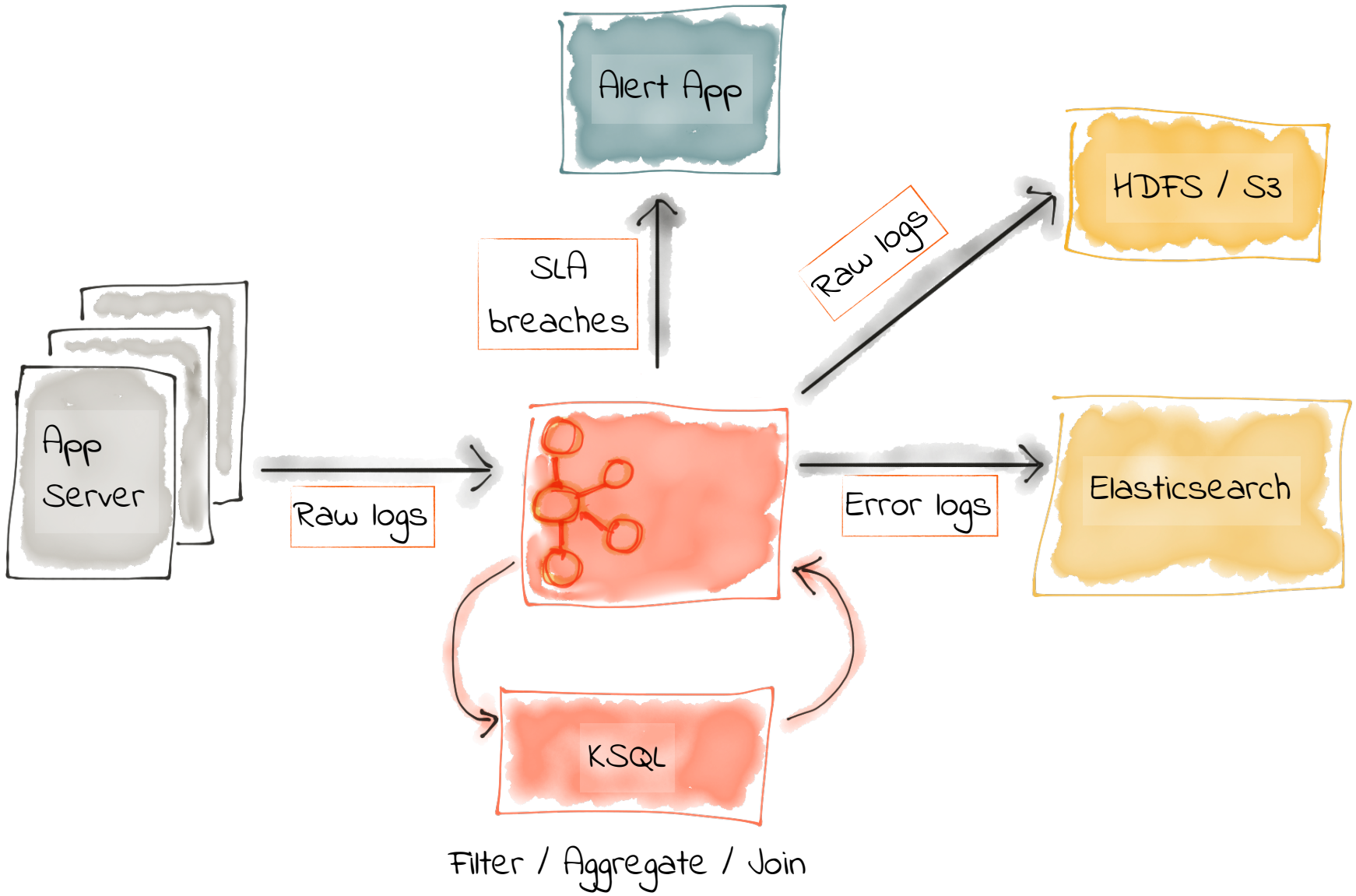
Streaming Transformations with KSQL



Filtering streams with KSQL

```
ksql> CREATE STREAM ERROR_LOGS AS
    SELECT * FROM LOGS
    WHERE RESPONSE >=400;

 Message
----------------------------
 Stream created and running
----------------------------
```

Streaming Transformations with KSQL



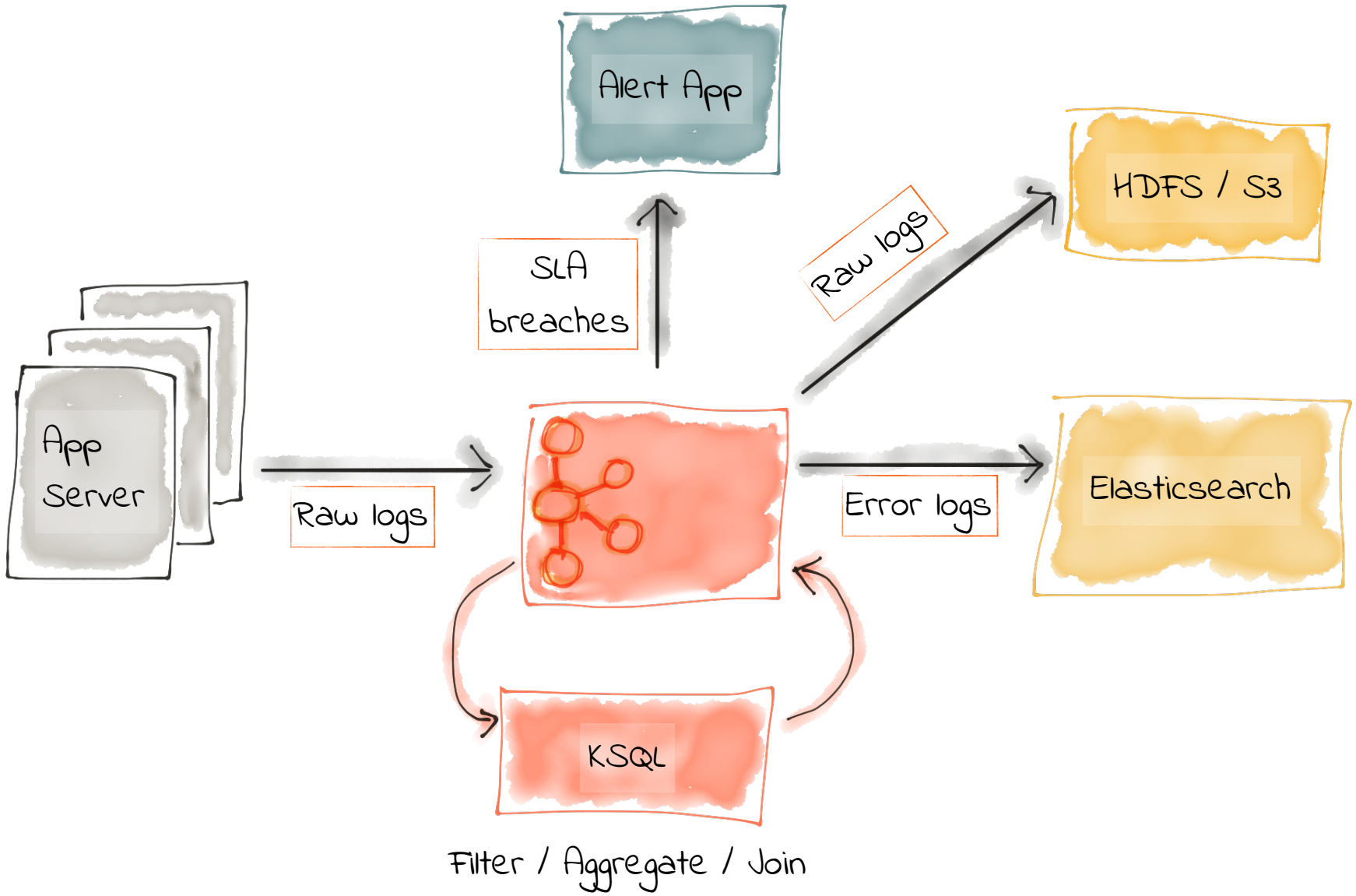
Monitoring thresholds with KSQL

```
ksql> CREATE TABLE SLA_BREACHES AS
      SELECT RESPONSE, COUNT(*) AS REQUEST_COUNT
      FROM LOGS
      WINDOW TUMBLING (SIZE 1 MINUTE)
      WHERE RESPONSE>=400
      GROUP BY RESPONSE
      HAVING COUNT(*) > 10;
```

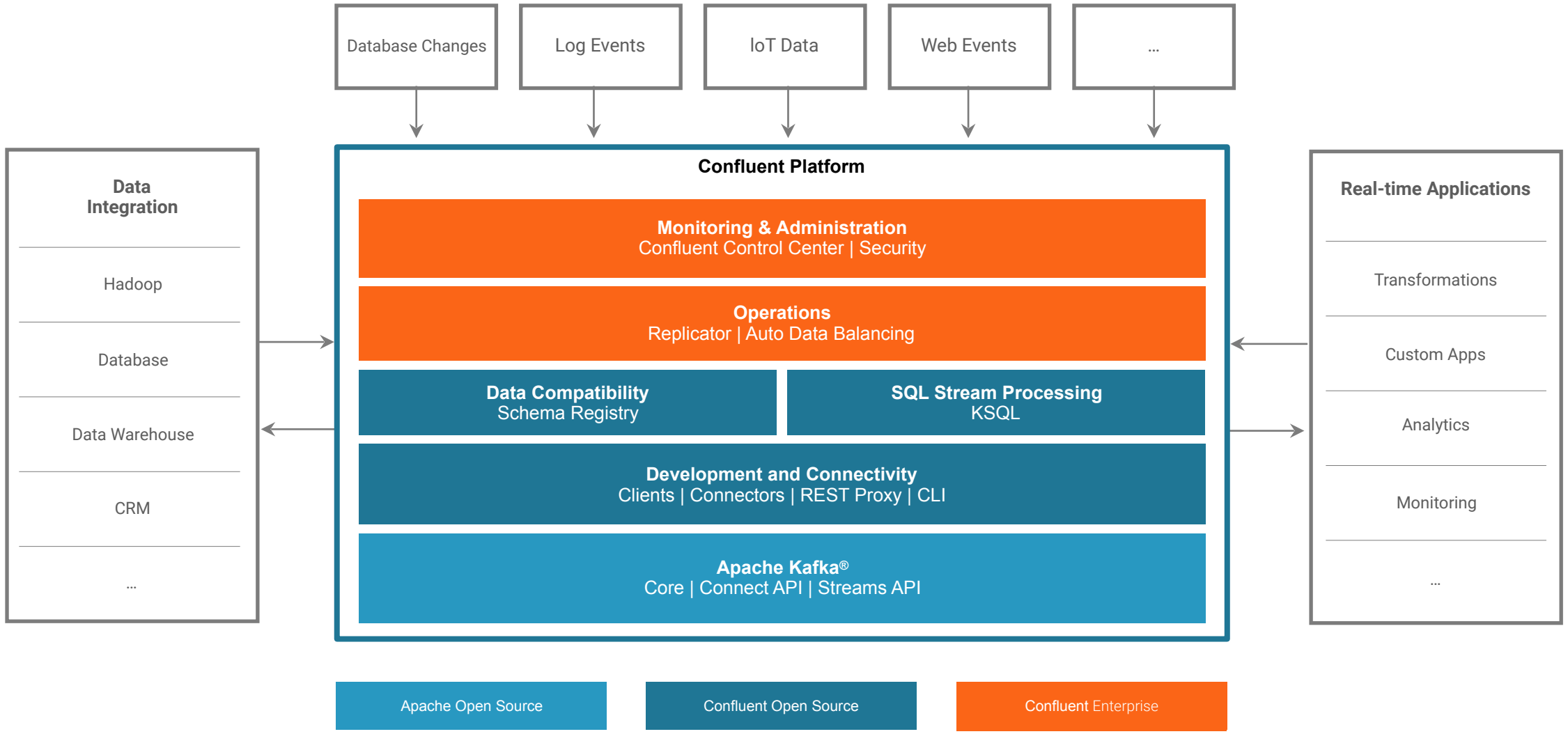
Monitoring thresholds with KSQL

```
ksql> SELECT TIMESTAMPTOSTRING(ROWTIME,
        'yyyy-MM-dd HH:mm:ss'),
     RESPONSE, REQUEST_COUNT
     FROM SLA_BREACHES;
2018-01-23 19:05:00 | 503 | 20
2018-01-23 19:06:00 | 503 | 31
2018-01-23 19:07:00 | 503 | 14
2018-01-23 19:08:00 | 503 | 50
```

Streaming Transformations with KSQL



Confluent Platform: Enterprise Streaming based on Apache Kafka®



kafka summit
25% Discount code!
KSE18Meetup
Kafka Summit 2018
Kafka Summit London: April 23-24, 2018
Kafka Summit San Francisco: October 16-17, 2018
www.kafka-summit.org
Presented by confluent

Streaming ETL, powered by Apache Kafka and Confluent Platform

https://www.confluent.io/blog/simplest-useful-kafka-connect-data-pipeline-world-thereabouts-part-1/
https://docs.confluent.io/current/connect/connect-elasticsearch/docs/

https://www.confluent.io/download/

Kafka Summit discount code! **KSE18Meetup**

@rmoff robin@confluent.io
https://speakerdeck.com/rmoff